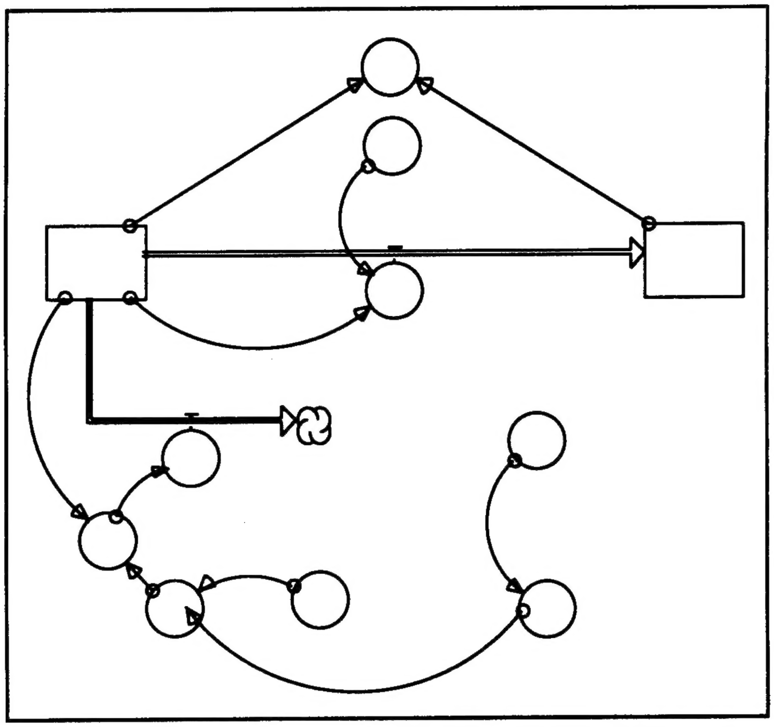
US Army Corps
of Engineers
Construction Engineering
Research Laboratories

USACERL Technical Report 95/16
August 1995

# Dynamic, Spatial, Ecological Modeling:

## A Demonstrated Simulation of the Sage Grouse Habitat at the Yakima Training Center, Washington

by
James D. Westervelt
Bruce Hannon
Pervaze Sheikh
Albert Cheng
Cory Rubin
Bruce Dvorak
Alicia Nugteren
Lynne Gildensoph
Helena Mitasova
Eric Lambert
Louis Iverson
Kenneth Pabich
Michael Shapiro



The U.S. Army is responsible for managing 12.4 million acres of land on 186 major installations worldwide. Proper land management supports the military mission and multiple use activities, but also presents the Army with a unique challenge as public land steward.

The U.S. Army Construction Engineering Research Laboratories (USACERL) is currently conducting basic research into alternative ways of managing limited military training lands through computer simulation of the interaction between natural processes and military training. Modeling and simulation provide an approach for experimenting with pieces of the physical world through conceptual representations. Ecologists have used modeling in recent decades to simulate system behavior, most often at the full ecosystem level. However, less recognition has been given to the importance of the spatial arrangements of ecosystem components.

This study created a working dynamic spatial model of a selected ecosystem using a cellular approach. A landscape was divided into regular cells. A single ecosystem model was developed with STELLA, and then applied simultaneously to each cell. The model was demonstrated using data on the Sage Grouse at the Yakima Training Center, WA. This exercise demonstrated the potential effectiveness of a suite of software capabilities designed to facilitate landscape design, development, and simulation.

The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products. The findings of this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

*DESTROY THIS REPORT WHEN IT IS NO LONGER NEEDED*

*DO NOT RETURN IT TO THE ORIGINATOR*

## USER EVALUATION OF REPORT

REFERENCE: USACERL Technical Report 95/16, *Dynamic, Spatial, Ecological Modeling*

Please take a few minutes to answer the questions below, tear out this sheet, and return it to USACERL. As user of this report, your customer comments will provide USACERL with information essential for improving future reports.

1. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which report will be used.)

________________________________________

________________________________________

________________________________________

2. How, specifically, is the report being used? (Information source, design data or procedure, management procedure, source of ideas, etc.)

________________________________________

________________________________________

3. Has the information in this report led to any quantitative savings as far as manhours/contract dollars saved, operating costs avoided, efficiencies achieved, etc.? If so, please elaborate.

________________________________________

________________________________________

4. What is your evaluation of this report in the following areas?

a. Presentation: ____________________________

b. Completeness:____________________________

c. Easy to Understand: ________________________

d. Easy to Implement: ________________________

e. Adequate Reference Material: ____________________

f. Relates to Area of Interest: ____________________

g. Did the report meet your expectations? ________________

h. Does the report raise unanswered questions? ______________

i. General Comments. (Indicate what you think should be changed to make this report and future reports of this type more responsive to your needs, more usable, improve readability, etc.)

______________________________________________

______________________________________________

______________________________________________

______________________________________________

______________________________________________

______________________________________________

5. If you would like to be contacted by the personnel who prepared this report to raise specific questions or discuss the topic, please fill in the following information.

Name: ______________________

Telephone Number: ______________________

Organization Address: ______________________

______________________

______________________

6. Please mail the completed form to:

Department of the Army  
CONSTRUCTION ENGINEERING RESEARCH LABORATORIES  
ATTN: CECER-TR-I  
P.O. Box 9005  
Champaign, IL 61826-9005

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*

| | | |
|---|---|---|
| 1. AGENCY USE ONLY *(Leave Blank)* | 2. REPORT DATE<br>August 1995 | 3. REPORT TYPE AND DATES COVERED<br>Final |

4. TITLE AND SUBTITLE
Dynamic, Spatial, Ecological Modeling: A Demonstrated Simulation of the Sage Grouse Habitat at the Yakima Training Center, Washington

5. FUNDING NUMBERS
4A161102
AH68
CT-IJ3

6. AUTHOR(S)
James D. Westervelt, Bruce Hannon, Pervaze Sheikh, Albert Cheng, Cory Rubin, Bruce Dvorak, Alicia Nugteren, Lynne Gildensoph, Helena Mitasova, Eric Lambert, Louis Iverson, Kenneth Pabich, and Michael Shapiro

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Construction Engineering Research Laboratories (USACERL)
P.O. Box 9005
Champaign, IL 61826-9005

8. PERFORMING ORGANIZATION REPORT NUMBER
TR 95/16

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Directorate of Research and Development
ATTN: CERD-ZA
20 Massachussetts Ave., NW.
Washington, DC 20314-1000

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Copies are available from the National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The U.S. Army is responsible for managing 12.4 million acres of land on 186 major installations worldwide. Proper land management supports the military mission and multiple use activities, but also presents the Army with a unique challenge as public land steward.

The U.S. Army Construction Engineering Research Laboratories (USACERL) is currently conducting basic research into alternative ways of managing limited military training lands through computer simulation of the interaction between natural processes and military training. Modeling and simulation provide an approach for experimenting with pieces of the physical world through conceptual representations. Ecologists have used modeling in recent decades to simulate system behavior, most often at the full ecosystem level. However, less recognition has been given to the importance of the spatial arrangements of ecosystem components.

This study created a working dynamic spatial model of a selected ecosystem using a cellular approach. A landscape was divided into regular cells. A single ecosystem model was developed with STELLA, and then applied simultaneously to each cell. The model was demonstrated using data on the Sage Grouse at the Yakima Training Center, WA. This exercise demonstrated the potential effectiveness of a suite of software capabilities designed to facilitate landscape design, development, and simulation.

14. SUBJECT TERMS
ecosystem
land management
military installation
simulation modeling

15. NUMBER OF PAGES
102

16. PRICE CODE

| | | | |
|---|---|---|---|
| 17. SECURITY CLASSIFICATION OF REPORT<br>Unclassified | 18. SECURITY CLASSIFICATION OF THIS PAGE<br>Unclassified | 19. SECURITY CLASSIFICATION OF ABSTRACT<br>Unclassified | 20. LIMITATION OF ABSTRACT<br>SAR |

# Foreword

This study was conducted for Headquarters, U.S. Army Corps of Engineers under Project 4A161102AH68, "Processes in Pollution Abatement Technology"; Work Unit CT-IJ3, "Dynamic, Spatial, Ecological Modeling."

The work was performed by the Natural Resources Assessment and Management Division (LL-N) of the Land Management Laboratory (LL), U.S. Army Construction Engineering Research Laboratories (USACERL). This work was also supported by the National Center for Supercomputing Applications (NCSA) at the University of Illinois, Urbana-Champaign (UIUC). Joseph Harding is an NCSA Associate Director and directs the software development group. Dr. Larry Smarr is Director of NCSA. Dr. Bruce Hannon is a professor in the Department of Geography at UIUC. Albert Cheng is a research programmer with NCSA. Louis Iverson, Lynne Gildensoph, and Helena Mitasova are research scientists, and Pervaze Sheikh, Alicia Nugteren, Cory Rubin, Bruce Dvorak, Eric Lambert, and Kenneth Pabich are graduate students at UIUC. Michael Shapiro is a USACERL research programmer. Dr. David J. Tazik is acting Chief, CECER-LL-N, Dr. William Severinghaus is Operations Chief, and William D. Goran is Chief, CECER-LL. The USACERL technical editor was William J. Wolfe, Technical Resources Center.

COL James T. Scott is Commander and Acting Director of USACERL, and Dr. Michael J. O'Connor is Technical Director.

# Contents

# List of Figures

## Figures

# 1 Introduction

## Background

Military land managers are often faced with extraordinarily difficult decisions, e.g., to determine how land is used, scheduled, rehabilitated. Such decisions must be made in light of multiple objectives that affect large areas of land over long spans of time. For example, land managers are responsible for:

- establishing satisfactory training of military personnel (DA 1979)
- sustaining the state of the land for support of training over decades (or even centuries)
- meeting legal requirements related to threatened and endangered species (TES), chemical spills and wastes, air and water pollution, and impacts of noise
- maintaining long-term viability of local ecosystems
- maintaining even longer-term effects on local and regional biodiversity
- providing appropriate opportunities for recreation, farming, grazing, timber harvest, and wildlife preservation
- maintaining aesthetic qualities.

Land managers must reconcile the needs and demands of advocates of these different and often conflicting objectives. Because it is virtually impossible to meet the demands of all advocates, the land manager is required to balance the objectives in coherent short, medium, and long-term management plans. Since the strongest local advocates are those assigned to train on the landscape, there is always a stronger impetus to meet the short-term requirements of the trainers. As a result, the requirements for long-term sustainability often result in reports demonstrating a general decrease in the ability of military lands to sustain recent training intensities (Diersing and Severinghaus 1984; Goran, Radke, and Severinghaus 1983; Johnson 1982; Schaeffer et al. 1986; Severinghaus and Goran 1981; Severinghaus, Riggins, and Goran 1979).

### *The Role of Scientific Studies*

Traditionally, the results of scientific inquiries go beyond the expressed interests of advocacy groups to provide: (1) the basis for the formal education of land managers, and (2) a permanent record and reference regarding land condition. This information better equips land managers to make the necessary and often difficult professional judgments demanded of them. Such judgments are actually based on models of the world that have been constructed through past and continuing education. Each new scientific investigation and report either validates or challenges that internal model; such challenges serve either to cause the new information to be discarded, or to adjust or replace the internal model.

### *The Role of Computer Models and Simulations*

Scientific studies and reports have a new role in the age of fast computer technology. The traditional application of scientific data results in powerful models of the environment in the minds of scientists, land managers, and laymen alike. As good as this traditional approach is, computer technology may be used to:

- *Reconcile different models.* Even with measurably identical training, every professional views (models) the world differently. The number of available "professional opinions" can always equal the number of professionals.
- *Overcome communication difficulties.* The perspectives created by different conceptions or models can make it very difficult for even "experts" to communicate effectively.
- *Resolve competition between different specialties.* Academic disciplines tend to fragment into many subdisciplines. Specialists can easily develop understandings (models) of the world that directly compete with each other. For example, different environmentalists may prefer to view an animal as an individual; a member of a population, species, or guild; as a part of a community or ecosystem; or as an assemblage of organ systems each responding to the chemistry of the environment.
- *To test internal models.* Internal models are difficult to challenge and defend. Senior scientists and managers are often presumed to have the best internal models and therefore the most defensible statements based on professional judgments. Until an internal model is formalized, it cannot be inspected or verified.
- *To foster interdisciplinary study.* Internal models are the result of individual training and experience. Very few individuals can master more than one discipline—let alone the myriad of disciplines ranging from military science, psychology, ecology and biodiversity, medicine, toxicology, and others—that have some knowledge, models, and views on the processes occurring within ecosystems and landscapes.
- *To clarify difficult dynamics.* Internal models or understandings are often inadequate for describing relationships more complex than direct cause-effect associations. Strings of cause-effects resulting is serious indirect relationships can be difficult for the human mind to comprehend and visualize.

- *To demonstrate complex spatial relationships.* Metapopulations exist in nature at all scales of space. Visualizing simultaneously the ebbs and flows of population densities of different species occurring at different speeds at different spatial scales is very difficult.

Computer simulations promise to address some of these concerns. Formal models (like formal journal articles) are unambiguous; they allow individuals to communicate more precisely, more completely, and more efficiently. Simulation models provide an environment for establishing connections between disparate pieces of scientific information, studies, and reports. Numerous authors representing different specialties can participate in the construction of formal models, which can then be reviewed and challenged. The process allows the models and the underlying scientific knowledge to improve. Most importantly, dynamic simulation models provide an environment for experimenting with and understanding indirect cause-effect relationships.

While landscape and spatially explicit ecological simulation is not a panacea (such models are no better than the data that underlie their structures), exploratory models might help scientists to discover some new information. Generally, errors in such models only reflect gaps in the knowledge on which the model is based. This is not a problem with modeling, for decisions are universally based on internal models of systems and nature. In fact, the formal capturing of such internal models opens them up for recognizing errors that might otherwise go undetected. An initial computer model simulation will begin to help land managers to translate internalized models of the environment into structured entities by formalizing information from an array of disciplines into a prototype software simulation model.

## Objectives

The objectives of this study were to:

1. Demonstrate the potentials offered by cellular modeling. This spatially-explicit dynamic approach to simulation provides ecologists and land-managers with a tool to test understandings of the processes that drive variations in spatial distributions of populations.
2. Develop software, hardware, management, and interdisciplinary skills required to create such models

## Approach

1. A multidisciplinary group of researchers was coordinated to define and resolve the problem of developing a cellular approach to modeling a dynamic ecosystem
2. Hardware and software environments for the model were specified.
3. A STELLA conversion program was developed to convert STELLA code into C for processing in a parallel processing environment.
4. The cellular model was developed in STELLA, and further software was developed to allow the output of the translator to be compiled in several hardware environments, including a SUN workstation, a network of SUN workstations, the CM-5 parallel processor, and a small network of transputers attached to a desktop Macintosh computer.
5. FORTRAN code was developed to allow movement of relevant information between individual cells.
6. Cellular modeling was conducted as an extension of the Geographic Resource Analysis Support System (GRASS) raster Geographic Information System (GIS). The output of cellular simulation took the form of digital maps that could be fed back through the GIS for further analysis and display.
7. Project performance, and the technical capabilities of the hardware and software configurations were evaluated, and recommendations were made for continued system development.

## Scope

This project was intended to form the foundation for the design and development of more comprehensive dynamic-spatial ecological models. The models created in this study were applied to real-world problems for demonstration only; they are not yet ready for incorporation into actual land management policy decisions. However, the software and hardware configurations used in this demonstration may currently serve to generate location, user, and land-use specific dynamic models.

Note that the model described in this report was intended to provide a realistic, but not real, example of the cell-based spatially-explicit modeling approach. The realistic "feeling" of the model was the result of a programming effort to help ecologists and land managers visualize how they might capture their knowledge and concepts.

## Mode of Technology Transfer

This demonstration project will form the foundation for the design of more powerful and flexible computer software.

# 2 Project Development

## Hardware and Software for Modeling

### *Hardware*

The hardware environments included color Apple Macintosh machines, UNIX-based workstations, and a CM-5 Connection Machine (chosen for its parallel processing capability). University of Illinois students and faculty used the Macintosh machines to design and develop the cellular ecological model. The UNIX environments were used to generate system-starting parameters in the form of digital maps and to develop the CM-5 software. The CM-5 was the target machine for running the simulations.

### *Software*

Software environments consisted of a combination of commercial, public domain, and special purpose programs. STELLA provided the basic modeling environment used by the multidisciplinary team. (Appendix A gives a brief description of the STELLA environment.) Conversion programs were developed to transform the STELLA models into C code that could be run in the CM-5 parallel processing environment. The public domain GRASS program was used to seed the model with a starting point. Specially written FORTRAN code allowed intercellular movement of information between adjacent grid cells.

***STELLA Translator.*** The cellular model was created in STELLA, and was simultaneously applied to all grid cells in the study area (342 cells x 342 cells = 116964 cells). The cell size of 30 meters square was chosen to represent an area that was normally large enough to hold only one female Grouse at any instant. The translation from the STELLA equations was done with software developed for this purpose (Maxwell 1993). The output of the translator can be compiled to run in several hardware environments including a SUN workstation, a network of SUN workstations, the parallel CM-5 machine, or a small network of transputers attached to a desktop Macintosh. For this study, the program was run on the CM-5 at a rate of 60 gigaflops per model year.

***FORTRAN Programming.*** The cellular model (created through STELLA) as applied in parallel (through Maxwell's translator) was not sufficient to generate a complete model.

Although the cellular model simulates the interactions within each cell, it does not provide rules for exchanging information between cells. A separate program written in FORTRAN was developed to provide movement of relevant information (movement of individual animals) between cells.

***GRASS.*** Cellular modeling is an extension of raster GIS technology. As in any simulation process, the starting state must be modeled and provided for the simulation. In the case of cellular modeling, the initial state for all cells was represented as a series of maps; each map provided a single-state variable for each cell. Cellular simulation output was also in the form of digital maps that could be fed back into the public domain GRASS (raster GIS) program for further analysis and display. Appendix B describes the GIS analysis steps used to generate impact maps that describe the result of a single type of training activity.

## Problem Definition

Researchers chose to model a problem provided by the Yakima Training Center (YTC), WA: the interaction between human training and Sage Grouse behavior in a threatened Sage Grouse habitat at YTC. Based on the recommendations of the YTC field biologist, a set of training ranges, home to a significant percentage of the state's threatened Sage Grouse population, was chosen. Military training landscapes are managed differently than the lands contiguous to and surrounding YTC. While neighboring lands are often highly managed for agriculture or human habitation, training land is likely to have been maintained in a more natural state. At YTC, a small community of Sage Grouse remains on a remnant of an original desert-steppe habitat that once extended over much of the Northwest. Although the installation is a more accommodating habitat to the birds than the surrounding private land, it was felt that this threatened species might better tolerate increased training frequency and intensity at YTC if training schedules were coordinated with seasonal variations in the sensitivity of the birds (e.g., nesting habits).

The Sage Grouse problem provided an excellent test ground for modeling capabilities for the following reasons:

- *It was spatial in nature.*—The problem incorporated the movement of individual Sage Grouse across the landscape.
- *It involved human activity.*—Because human activity is involved, the problem must consider management and policy decisions, making it more interesting to communities that benefit directly from the development of ecological models.

- *It represented a real problem.*—Although theoretical problems can be effective test environments, the specification of a real problem by real land managers provides effective additional motivation. However, the results here are insufficient for the management of training.
- *It involved nonmovable components.*—This problem required the modeling of sessile communities such as sagebrush, forbs, and grasses based on physical qualities such as slope, elevation, aspect, and soil characteristics.
- *Data was readily available.*—Because the research was largely unfunded, extensive data collection was impractical. YTC has a good spatial database and had recently completed documentation on the study of the Sage Grouse communities. Such data is based on field studies that provide necessary input for model design and calibration. Data sources are returned throughout later discussions of the model.
- *Land managers already involved in the problem were interested in finding a solution.*—The Sage Grouse provided a problem important to environmental groups and to Army installations. There are strong interests in managing the land exclusively for the Grouse and equally strong requirements to use the land exclusively for military training. The result, while only a demonstration of the practicality of the process, was an important focus for policy debate and land management innovations.
- *Researchers saw the problem as an interesting challenge.*—This problem was an example of the kind of modeling that provides significant academic challenges. It simultaneously requires degrees of simple cause-effect modeling with spatial behavior modeling and also involves chemical, physical, and biological processes.
- *The problem represented a real-world exercise for students*—Technology used in this project will be used by today's students as they enter the world of land management. An additional positive project outcome was that it trained students in the use of modeling techniques in future research and applications.

## Project Management

From a management perspective, it would have been impossible for the 20-person modeling group to be responsible for every aspect of the model. For this reason, four modeling groups (Sage Grouse, Vegetation, Human, and Physical) and a software development group were formed. The software group developed the software that allowed the cellular model created by the other groups to be run on the CM-5 Connection Machine. It was also responsible for designing the algorithm to provide for the intercellular movement of the Sage Grouse.

The STELLA software is an excellent organizing device for the design and development of single models, written either by individuals or by groups. In this case, four separate groups developed different parts of the final model. Therefore, a shared base model was created to provide a simplified time-series of anticipated output from the other models within which individual submodels could be developed. The base model contained the expected output of each of the four groups' efforts. Each group continually updated the simplified section of the base model that held the place for their work with the submodels they developed. This approach provided a common ground for communication and a straightforward approach to combining group efforts.

The four subgroups spent most of one university semester developing sections of the final model. During this time, the class assembled once a week to present the status of their individual work. Communications between the groups identified successes, failures, and specific group needs for connecting the separate submodels.

Intercellular movement (discussed in more detail in Chapter 4, p 48) of the Sage Grouse was facilitated by research activities outside of the class cellular modeling effort. This effort was key to meeting the goal of modeling the effect of spatial location and distribution of the birds with respect to training activities.

While the class developed a cellular model based on a range of expected system starting points, the groups understood that the initializing factors for each grid cell would come from a "snapshot" of the system represented by a set of digital maps prepared within and stored by a GIS. Again, these maps were developed by outside researchers.

Once the cellular model was completed, the migration algorithm designed and created, and the initializing GIS data generated, these components were brought together. The combination required debugging and simplification before any results could be obtained.

The debugging process required a significant amount of model simplification. The model generated by the group had a combination of short-term (week-oriented) impacts and long-term (year-oriented) impacts. The extracted short-term impact section of the model was debugged and demonstrated. This reduced model had neglected, for example, the impacts on the Grouse via tracked vehicle compression of the soil, and on the sage brush community via tracked vehicle damage (long-term impacts).

Once a demonstrable output could be created with the reduced model, the project had reached its goal: to use a real problem affecting real people with real land management problems to demonstrate the technical capabilities of a specific configuration of hardware and software tools in the hands of a multidisciplinary group of researchers. With sufficient interest, field work, and participation by end-user land managers, such models could become a powerful tool for land management and endangered habitat protection.

This model could have been calibrated to some degree if a more detailed specification of the current and historic training patterns was available. For obvious reasons, such schedules are classified. Such data would allow construction of a long-run model to determine the expected current total grouse population and to compare it with the results of known surveys. Instead, results of the following scenarios were computed: full training, limited training, and no training. The general spatial reaction of the grouse and their reported total number were then given.

The reader must keep in mind this report discusses the spatial modeling process from a demonstration point of view. It was necessary to determine whether sufficient data exists for a team of researchers to be profitably employed in developing a simulated, spatially explicit, dynamic record of a specific species and its reaction to differing levels of human intervention with it and its supporting ecosystem. The tentative conclusion is that the "critical mass" *does* exist, for a class of students was able to develop a reasonable demonstration model in a single semester's work.

# 3 The Full Model

This chapter describes the STELLA submodel components and the design and development of the intercellular migration algorithm. Appendix C provides a full listing of the equations described here.

Figure 1 shows the complete model initially generated by the researchers. Although the details are not visible, the pieces discussed in the following subsections can be visually mapped back to this figure. Starting from the top left and moving clockwise, the boxes contain:

1. Base model (All other components communicate through this common section.)
2. Physical submodel
3. Human submodel
4. Vegetation submodel
5. Grouse attractiveness submodel
6. Female Sage Grouse model
7. Extra CM-5 input variables.

Appendix C, which contains the STELLA equations and internal system documentation, gives more details on the cellular model.

## Sage Grouse Section

The core requirement of this exercise was to simulate the impact of military training on the Sage Grouse, so the life cycle of the Sage Grouse on YTC was modeled along with the factors that influence Grouse survival (physical environment, vegetation, and human impact). Inputs required for the processing of this model included sagebrush density (from the vegetation section), the noise index (from the human section), and both snow cover and temperature (from the physical section).

The initial distribution of the Sage Grouse consisted of 200 female Grouse in the 116,964 30x30m grid cell area. Because the densities of these animals was so low, it was unreasonable to model entire population densities (as opposed to individuals). At the spatial resolution of $30m^2$ cells, the average Grouse density is approximately 0.0017

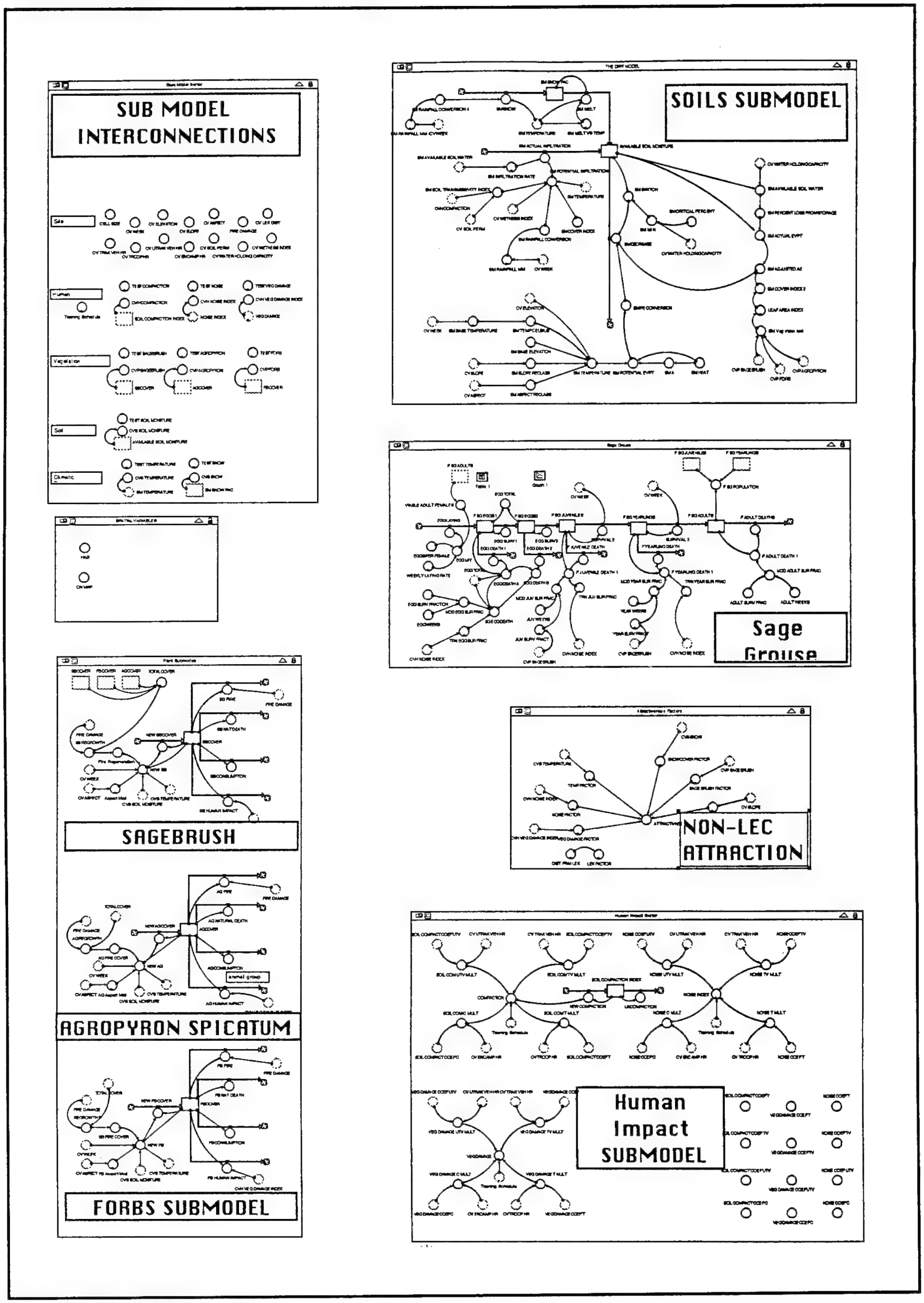


**Figure 1. Full model.**

birds per cell. These 200 female Grouse were distributed using the GRASS random command across the regions of the training area that had higher densities of sagebrush.

For this model, only female Grouse were considered; the abundance of males was presumed not to be a limiting factor within the anticipated range of conditions. Very few males are required to ensure fertilization of the females. Adequate female care of the young was probably the current limiting factor in survival of the population. In addition, this model allowed only one female Grouse, plus eggs and associated juveniles, to exist per cell. All stocks in the Grouse submodel (egg, juvenile, and adult populations) were established as integers. The total average life span of the Grouse used in this model was 5 years. The review presented is sectioned according to the life cycle stages of the Sage Grouse, beginning with mating adults and fertilization of the females (Figure 2).

### *Fertilization*

Female Sage Grouse are called to the lek and then fertilized there by males during the mating season. (A lek is an area of land where mating occurs seasonally on a consistent basis.) The number of males on the lek was estimated by a graph that relates males on the lek to the week of the year and that is contained in MALES_ON_LEK (Figure 2). This graph showed the 12 weeks when Grouse are active on the lek for mating. CV_LEK_DIST measured the distance (in meters) of Grouse from the nearest lek. This distance was important for identifying the direction of the lek from the female positions on the range and for indicating when a female is on the lek to mate. The number of female and juvenile Sage Grouse provided additional input to FERTILIZE. This input and their origin are discussed later in this section. The final equation for fertilization was :

```
FERTILIZE = IF (CV_LEK_DIST = 0 & F_SG_PREGNANT=0 AND                    [Eq 1]
F_SG_JUVENILES=0) THEN F_SG_ADULTS * MALES_IN_LEK * 0.1 ELSE 0
```

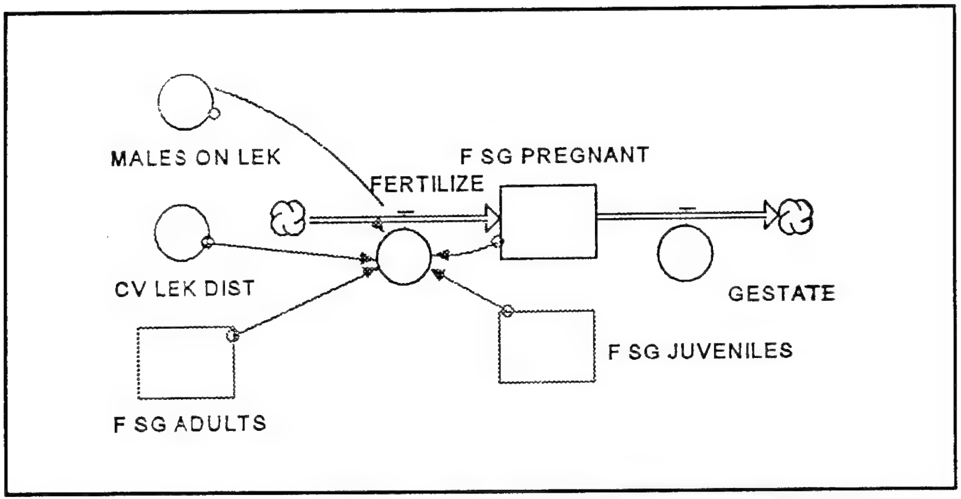


Figure 2. The mating and fertilization section of the Sage Grouse model.

The variable FERTILIZE was added to F_SG_PREGNANT to give the number of pregnant Sage Grouse. This number was added to GESTATE, which was set at one DT (the time it takes to perform all the commands in the program). For this model, the DT was set at 1 week.

### *Egg Laying Capacity*

Since only one female was assumed to exist per cell, egg-laying capacity was modeled for each individual. Sage Grouse lay between 6 to 10 eggs per clutch (Taffe-Pounds 1992). An estimate of six eggs per clutch was used in the model and was recorded in EGGS_PER_FEMALE. This was converted to EGG_LAY, which determined if hatching occurred in the cell, a condition based on the fertility condition of the resident female in that cell.

```
EGG_LAY= IF (FERTILIZE≠0) THEN EGGS_PER_FEMALE ELSE 0                    [Eq 2]
```

The value of EGG_LAY was then converted to a whole number in the equation presented in Equation 3 by randomly rounding EGG_LAY up or down. The (1/DT) forced the full number of eggs to be laid for any chosen time step. The (+ 1 ) statement in the equation allowed the whole number to be rounded up to a full unit or down to the base unit.

```
EGG LAYING =                                                             [Eq 3]
IF (RANDOM (0.0,1.0) < (EGG_LAY - INT (EGG_LAY)))
THEN (1/DT) * (INT (EGG_LAY) + 1) ELSE (1/DT) * (INT (EGG_LAY))
```

Eggs now need to survive two 1-week time steps. The whole number of eggs produced was stored in the state variable, F_SG_EGGS_1. The eggs that survived F_SG_EGGS_1 were randomly rounded to a whole number in the flow, EGG_SURV_1:

```
EGG_SURV_1 = IF (RANDOM (0.0,1.0) < ( DT * F_SG_EGGS_1 -                 [Eq 4]
INT (DT * F_SG_EGGS_1))) THEN (1/DT * DT )) * (INT (DT *
 F_SG_EGGS_1) + 1)  ELSE (1/ (DT * DT)) * (INT (DT * F_SG_EGGS_1))
```

The resultant value was then converted to F_SG_EGGS_2. The sum of both F_SG_EGGS_1 and _2 equaled EGGS_TOTAL.

Egg survival depended on the mortality pressure during each of the two 1-week time steps. (Figure 3).

### *Egg Mortality*

Egg mortality was calculated by a factor equated in SG_EGG_DEATH (eggs dying). This factor was derived from the EGG_SURV_FRACTION and the incubation

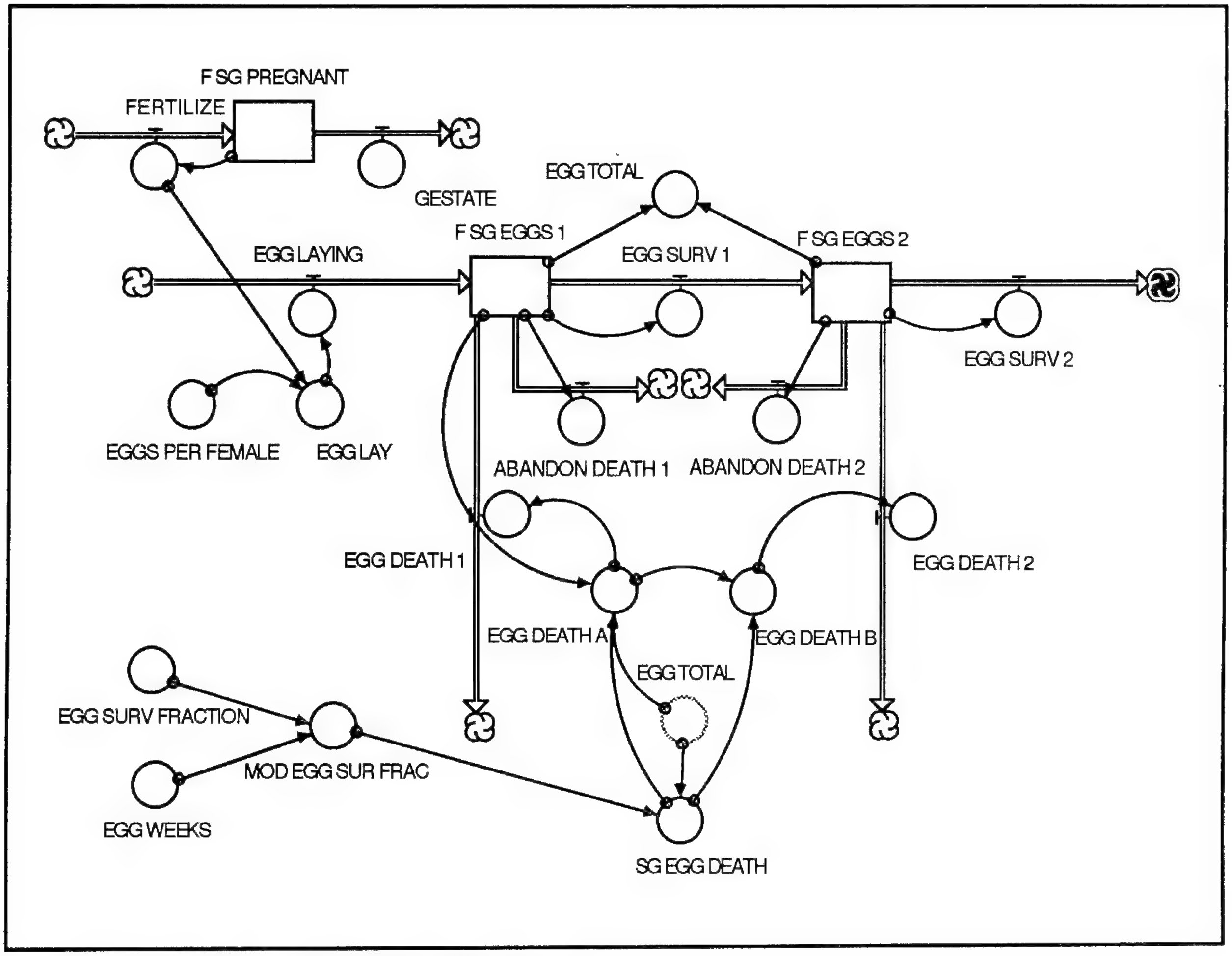


**Figure 3. The mortality and laying submodel for eggs.**

(EGG_WEEKS). EGG_SURV_FRACTION was the percentage of eggs surviving per clutch and is set at 38 percent (Eberhardt and Hofmann 1991). EGG_WEEKS was set at 2 weeks, the time needed for incubation to occur (Dalke et al. 1963). Both of these factors entered the converter, MOD_EGG_SUR_FRAC (model egg survival fraction). Factors causing death in Sage Grouse eggs were included:

MOD_EGG_SUR_FRAC = 1/DT * ( 1 - EXP( LOGN (EGG_SURV_FRACTION) *(DT/ EGG_WEEKS))) **[Eq 5]**

The survival fraction flowed into SG_EGG_DEATH and combined with the total number of eggs to determine egg mortality/DT:

SG_EGG_DEATH= DT *(1-MOD_EGG_SUR_FRAC) * EGG_TOTAL **[Eq 6]**

SG_DEATH included EGG_DEATH_A and EGG_DEATH_B. EGG_DEATH_A represented the fraction of eggs dying from F_SG_EGGS_1; EGG_DEATH_B represented eggs dying from F_SG_EGGS_2:

1. EGG_DEATH_A = IF ( EGG_TOTAL =0) THEN 0 ELSE ( F_SG_EGGS_1 / EGG_ TOTAL) * SG_EGG_DEATH [Eq 7]
2. EGG DEATH B = SG_EGG_DEATH - EGG_DEATH_A

EGG_DEATH_A was rounded to a whole number to give EGG_DEATH_1 and EGG_DEATH_B was rounded to get EGG_DEATH_2 (see Equation 4).
EGG_DEATH_1 counted the eggs dying from the stock of F_SG_EGGS_1.
EGG_DEATH_2 counted the dead eggs from F_SG_EGG_2.

### *Juveniles*

From F_SG_EGGS_2, eggs that hatched became juveniles through the flow EGG_SURV_2, and were rounded to a whole number (see Figure 4 and Equation 4). The number of juveniles was recorded in the stock F_SG_JUVENILES.

Juvenile death was calculated by the juvenile survival fraction (JUV_SURV_FRACT). One part of this graph was derived from sagebrush cover (CVP_SAGEBRUSH, from the vegetation section), which is essential for the survival of juveniles. (More cover means

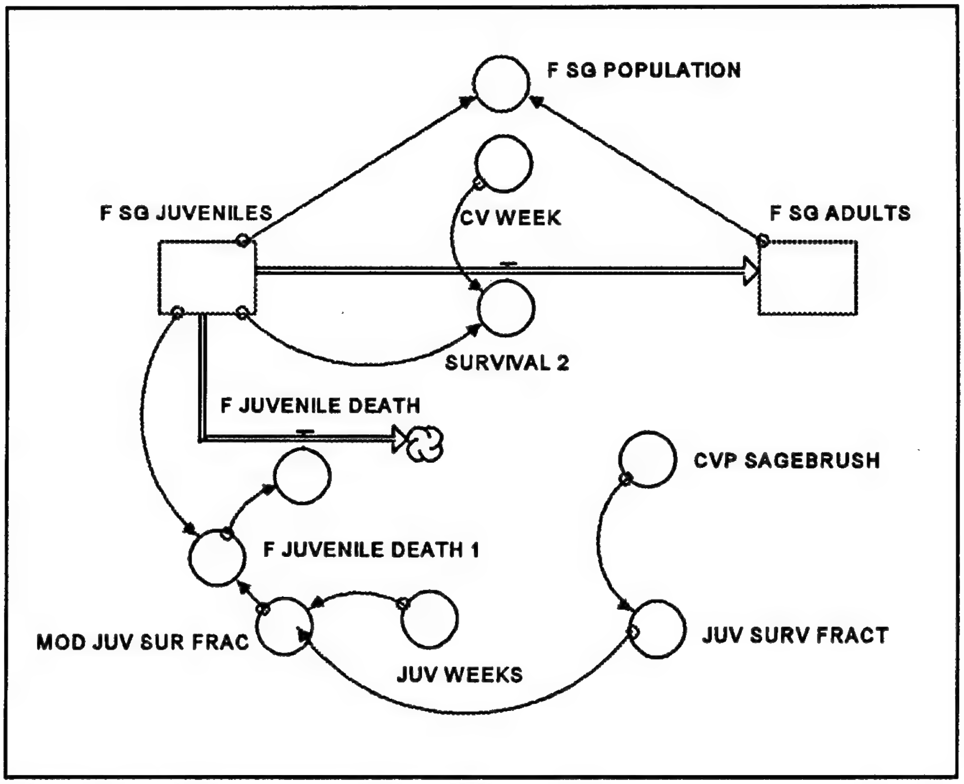


**Figure 4. The survival submodel for juveniles.**

less predation.) CVP_SAGEBRUSH was the percent of cover for the grid and was transformed with a graph into a survival fraction for juveniles. This fraction converted JUV_WEEKS (50 weeks to complete the first year) to give the fraction of juveniles to survive per time-step DT:

```
MOD JUV SUR FRAC = (EXP (LOGN ( JUV_SURV_FRAC) * (DT          [Eq 8]
/JUV_WEEKS)))
```

This fraction was incorporated to determine juvenile deaths in F_JUVENILE_DEATH_1:

```
F_JUVENILE_DEATH_1 = DT * ( 1- ( MOD_JUV_SUR_FRAC)) *          [Eq 9]
F_SG_JUVENILES
```

F_JUVENILE_DEATH_1 was converted into F_JUVENILE_DEATH (cf. Equation 4), giving the number of juveniles that die each DT. The surviving number of Grouse was transferred to the F_SG_ADULTS stock.

### *Adults*

SURVIVAL_2 took the number of surviving juveniles and transferred them to adults. In this equation, CV_WEEK allowed the stock of juveniles to graduate to adults every 52 weeks.

```
SURVIVAL_2= IF (CV_WEEK = 0) THEN F_SG_JUVENILES/DT ELSE 0          [Eq 10]
```

Adult survival was calculated with a survival fraction, set at 0.30 per DT, that accounted for factors that affect Sage Grouse survival. ADULT_SURV_FRACTION was combined with ADULT_WEEKS to give the MOD_ADULT_SUR_FRAC (Equation 8). This flowed into F_ADULT_DEATH_1 with F_SG_ADULTS, resulting in the number of deaths of adult Sage Grouse per DT (Equation 9). This value has been assigned into whole numbers of Grouse and subtracted from the store of F_SG_ADULTS. After the number of Grouse was calculated, the values of F_SG_ADULTS and F_SG_JUVENILES were combined to equal the stock, F_SG_POPULATION. This number of Sage Grouse were eligible for reproduction. The sum was transferred back to start the cycle again (Figure 5).

### *Sage Grouse Attractiveness*

The attraction model provided input (LEK_ATTRACTION, GENERAL_ATTRACTION, and DESIRE_TO_MIGRATE) for the migration algorithm. If males were booming on the lek (part of the mating ritual), attraction to it was recorded and captured by the LEK_ATTRACTION value. GENERAL_ATTRACTION provided a composite attraction value as a function of noise, snow, and cover.

DESIRE_TO_MIGRATE measured the responses of the Grouse to the factors affecting its decision to migrate to another cell. In addition to GENERAL_ATTRACTION, it was a function of BROOD_DESIRE, NESTING_DESIRE and the NOISE_FACTOR. Each factor compiled into the DESIRE_TO_MIGRATE flow will be discussed; concluding with their summation in the flow.

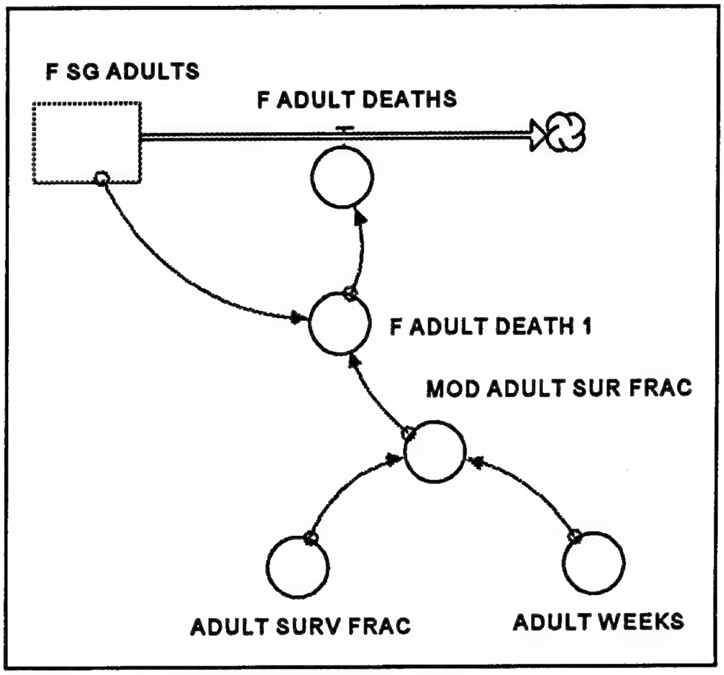


**Figure 5. The survival submodel for adults.**

Three variables determined the GENERAL_ATTRACTION of a cell to Sage Grouse: snow cover, sagebrush, and noise. Each was assigned a value ranging from zero to 1 (#1= most desirable value for Grouse, #0= least desirable). These values were plotted on conversion graphs contained in each factor unit (see Figure 6). The curves that are generated from these graphs are based on preferential parameters that give points to generate curves. A conversion graph took data from the indices shown (initialized through a GIS) and plotted it against another axis ranging from zero to 1. Values near 1 were assigned to preferential parameters (for the Sage Grouse) for the factor; values that were undesirable were assigned a value near zero. Preferential parameters are discussed in each factor section following. These values were multiplied together to create the GENERAL_ATTRACTION numeral:

GENERAL_ATTRACTION= SNOW_COVER_FACTOR * SAGE_BRUSH_FACTOR * NOISE_FACTOR **[Eq 11]**

Three factors affecting GENERAL_ATTRACTION, and some preferential parameters used to create the conversion graphs were:

1. The SAGEBRUSH_FACTOR measured the response of female Grouse to the percent cover of sagebrush as obtained from CVP_SAGEBRUSH. Braun et al. (1977) showed that at least 27 percent canopy cover was desirable for nesting. This percentage was used as an estimate in all models that included an attractiveness factor.

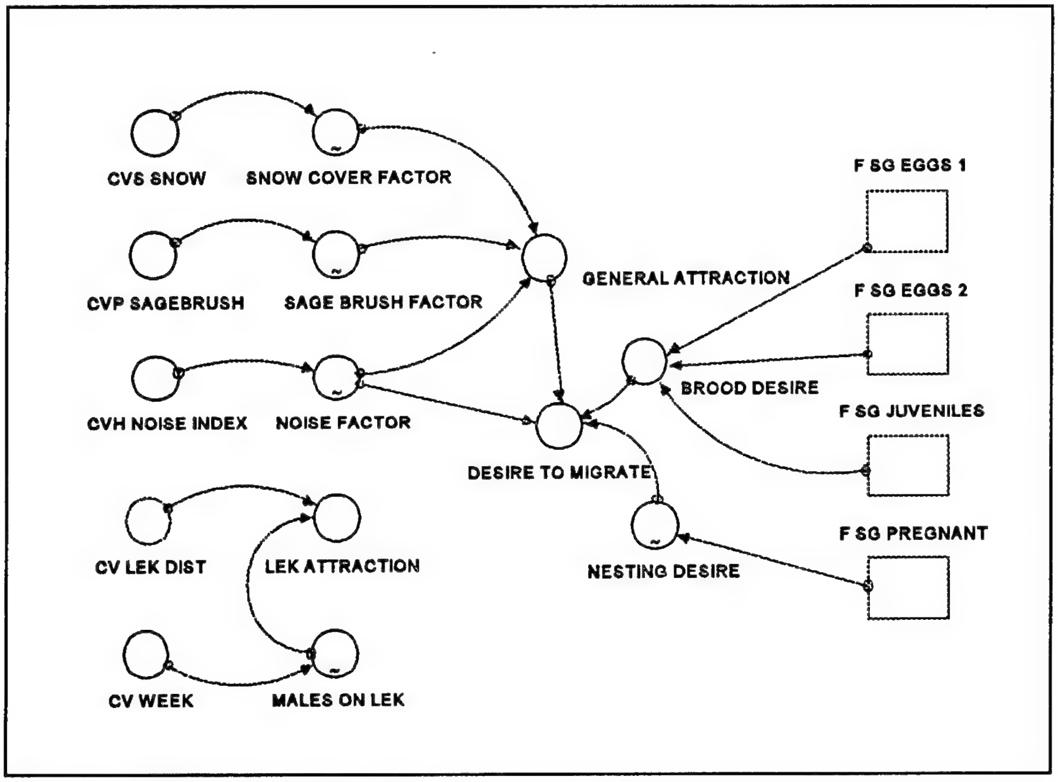


**Figure 6. The lek attractiveness model.**

2. The SNOW COVER_FACTOR analyzed the desirability of snowfall for the Grouse. Falls greater than 6 in. are unfavorable and trigger migration. This depth is also detrimental for foraging (Dalke 1963). Higher values had lower attractiveness ratings.
3. The NOISE_FACTOR estimated females' tolerance to noise from training maneuvers. This is discussed in "Human Impact" (p 42) and was derived from the CVH_NOISE_INDEX.

The BROOD_DESIRE flow measured the desire to stay in a cell based on maternal instincts. This flow rating ranged from zero to 1 and consisted of F_SG_EGGS_1, F_SG_EGGS_2 or F_SG_JUVENILES. Females with eggs present assumed nesting habits and stayed in the cell:

```
BROOD_DESIRE= IF (F_SG_EGGS_1 + F_SG_EGGS_2 >0)                    [Eq 12]
THEN 0.1 ELSE
IF (F_SG_JUVENILES >0) THEN 0.4 ELSE 1.0
```

NESTING_DESIRE was a zero to 1 factor representing the female's desire to remain in her current position based on her state of gestation. As she approached the egg laying stage, the desire to be mobile decreased. This factor was calculated from F_SG_PREGNANT. The graph is made according to time of pregnancy and ability to maneuver:

NESTING_DESIRE= GRAPH(F_SG_PREGNANT) **[Eq 13]**

The noise factor in this equation was the same as described in GENERAL_ATTRACTIVENESS and was applicable in mating and other scenarios. The final equation for DESIRE_TO_MIGRATE was:

DESIRE_TO_MIGRATE= ( 1- GENERAL_ATTRACTION) * MAX(0.0, NESTING_DESIRE* BROOD_DESIRE - NOISE_FACTOR) **[Eq 14]**

A factor near zero from this equation indicated no desire to migrate; a value near 1 showed the opposite.

Another factor determining attractiveness of an adjacent cell was LEK_ATTRACTION. Modeled separately, it indicated attractiveness increased as the noise from the lek increased; which itself is simply presumed to be a linear function of the number of males on the lek and the distance to the lek. The distance from the lek was recorded in CV_LEK_DIST for each female. During the mating season, the cells closer to the lek had higher ratings; conversely, Grouse further from the leks had a lower rating. This rating measured the attraction for female Grouse to the leks for mating. MALES_ON_LEK was the actual number of males on the lek, based on the time of the year and on experimental findings. MALES_ON_LEK and CV_LEK_DIST combined to form the LEK_ATTRACTION value:

LEK_ATTRACTION = CV_LEK_DIST * MALES_ON_LEK **[Eq 15]**

A higher value attracted female Grouse to the lek and a lower one discouraged them, signifying that females were looking for nesting sites after copulation.

## Vegetation Section

The submodels for the three vegetation types that occur at YTC were: sagebrush, grasses, and forbs. The cover of these types on the study site is 53 percent of the total cover on the site. Specifically, *Agropyron spicatum* (grass species), *Artemesia tridenta* (sagebrush species), and forbs represent 88 percent of the vegetative cover (Eberhardt and Hoffman 1991). Submodels represented the sagebrush, grass, and forb species; output for each submodel was expressed as the percent of total cover.

### *Sagebrush Submodel*

The main state variable was sagebrush cover (SB_COVER in Figure 7). This stock had several inputs and outputs. The initial SB_COVER was set via a raster GIS map derived from a satellite image. (Appendix D describes the GIS maps used in this demonstration.) SB_COVER will change over time from cover added due to the annual growth of certain plants, and from cover subtracted due to plant mortality. According to McArthur and Welch (1982), the death and growth in sagebrush communities cancel each other out, resulting in stable equilibrium over time. In this model, growth of the sagebrush community was driven by NEW_SB_COVER, which input added cover into SB_COVER. SB_COVER for each cell had a maximum capacity of 25 percent and was included in the equation for new sagebrush cover below (YTC Range Site Description 1989):

NEW SB COVER = MIN(NEW_SB, SB_COVER - 25) **[Eq 16]**

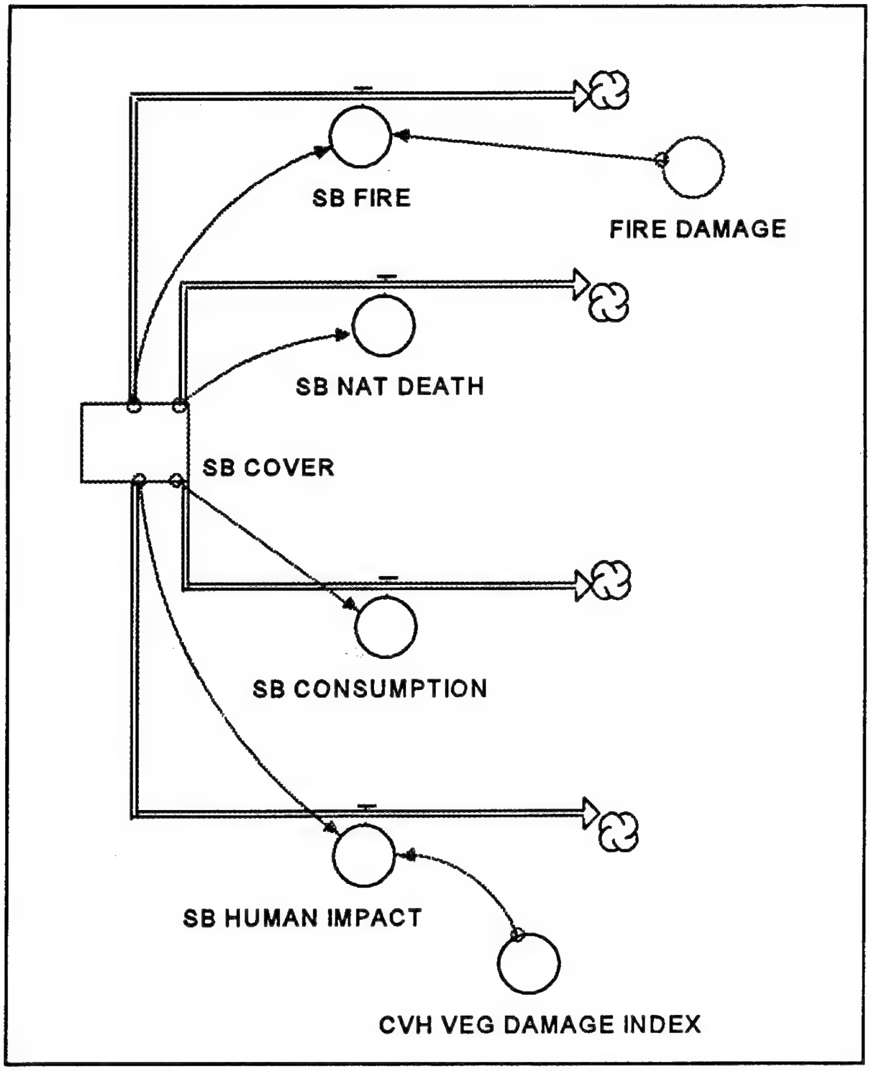


**Figure 7. Outflows for the sagebrush model (identical to other submodels).**

NEW_SB was a composite of variables relating to the environment during the time of year sagebrush grows based on data collected by Daubenmire (1975a):

```
NEW SB = IF (CVS_TEMPERATURE >12.0)                                    [Eq 17]
AND (CVS_TEMPERATURE<22.0)
AND (CVS_SOIL MOISTURE>8.0) AND (CVS_SOIL MOISTURE <10.0)
AND (CV_WEEK>15) AND (CV_WEEK<25)
THEN SB_COVER * (Aspect_Mod * 0.02) + Fire_Regeneration ELSE 0
```

Temperature was a function of a seasonal temperature and local elevation. Units were measured in degrees celsius. Equation 17 gave the ideal temperature at which sagebrush grows (12 to 22 °C). Soil moisture is measured at 15cm below the surface and was recorded as millimeters of water. Soil moisture values were taken from the physical section. Ideal soil moisture for sagebrush growth was modeled at a range between 8 and 10mm of water in the top 15cm of soil. The weeks of the year representing the growing stages for sagebrush were given in the equation. Secondary growth was modeled at 2 percent per year (McArthur and Welch 1982). The variable Fire_Regeneration monitored sagebrush regrowth after a fire. Full sagebrush regrowth can take up to a year and a half after a fire (Daubenmire, 1975a Miller et al. 1986) and can only regrow in a successional pattern after seeds are distributed on site by animals. This was incorporated in Equation 18 using a sagebrush regrowth flow:

```
Fire_Regeneration = DELAY ( SB_REGROWTH,78)                            [Eq 18]
```

This equation allowed 78 weeks to pass before regrowth could occur. Regrowth was calculated with:

```
RE GROWTH = IF FIRE DAMAGE = 1 THEN 0.05*TOTAL COVER ELSE 0            [Eq 19]
```

indicating that the regrowth percent cover after a burn for sagebrush was 5 percent of the total viable vegetative cover after 1 year and 7 percent after 2 years (Humphrey 1984). In TOTAL_COVER, the percent cover for the three types of vegetation was added together and a percent cover value was obtained. According to the YTC range site description (1989), total cover can reach a maximum of 59 percent of the area in each cell. Therefore, a homogenous stand can only reach 59 percent cover of the total area in any cell.

Aspect is the direction an object or group of objects face and was given through Aspect_Mod. Aspect_Mod had a graph that related azimuth in degrees to an index ranging from zero to 1. Azimuth measurements came from a GIS analysis of a digital elevation model (DEM) and were conveyed into STELLA by CV_ASPECT_SB. Sagebrush grows mostly on the south and west sides of hills and at elevations of 1520 to 2150m where soil moisture is sufficient (Barker and McKell 1983; Bonham et al. 1991). These locations were captured by the index.

Four outflows existed from SB_COVER (Figure 7). Each represented factors that depleted the percent cover of sagebrush in the system (Figure 8).

Fire in the system was regulated by FIRE_DAMAGE, which was obtained from the base model and was generated with GRASS data on fire occurrences. In the event of a fire, FIRE_DAMAGE was equal to zero; with no fire it was equal to 1. It was assumed that all sagebrush in a cell would be reduced to ash in a fire:

SB_FIRE = IF FIRE_ DAMAGE = 1 THEN SB_COVER ELSE 0 **[Eq 20]**

indicated that if a fire occurred (FIRE_DAMAGE = 0), all sagebrush vegetation would be burned. The next outflow reflected the normal dying of sagebrush plants over the year:

SB_NAT_DEATH = SB_COVER *.02/52 **[Eq 21]**

Since this was known to be a relatively stable community (West et al. 1979; McArthur and Welch 1982), natural death was modeled to reflect a 2 percent decrease in growth over 52 weeks in a year.

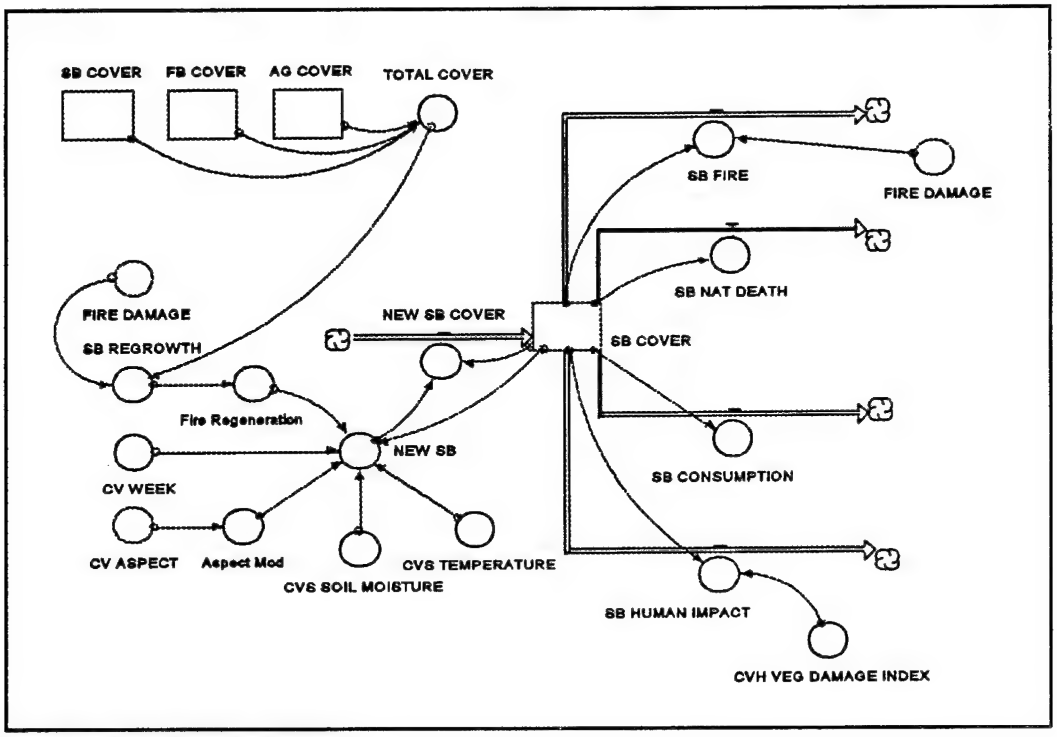


**Figure 8. Sagebrush submodel.**

Sagebrush consumption by animals (especially by the Sage Grouse) was the third outflow from this system. According to Taffe-Pounds (1992), this consumption was not considered to be a significant loss. Therefore, its parameter was set at zero, although it can be changed if future data indicates significantly higher consumption. The fourth and final outflow from the sagebrush system consisted of sagebrush loss due to human impact:

```
SB_HUMAN_IMPACT = IF ( CVH_VEG_DAMAGE_INDEX >0)                    [Eq 22]
THEN SB_COVER - (CVH_VEG_DAMAGE_INDEX * SB_COVER)
ELSE 0
```

Sagebrush cover was reduced by a factor of a vegetative damage index provided by the human section. This index related cover loss due to human activities.

### *The Agropyron Model (Grasses)*

The *Agropyron spicatum* model (Figure 9) approximated the growth and death of the major grasses at the YTC. The stand was assumed to be virgin because the YTC report had no listing for *Bromus tectorum*, a grass invader known to be successful after a fire and a potential competitor with *Agropyron*. This model was similar in structure to the sagebrush model, although some equation parameters were changed. For example, NEW_AG reflected growing conditions optimum for grasses:

```
NEW_AG = IF (CVS_TEMPERATURE>15)                                   [Eq 23]
AND (CVS_TEMPERATURE<30) AND (CV_WEEK 18) AND
(CV_WEEK 28) THEN AG_COVER * (AG_Aspect_Mod *0 .02)
+ (SOIL_MOISTURE*0) + AG_FIRE COVER ELSE 0
```

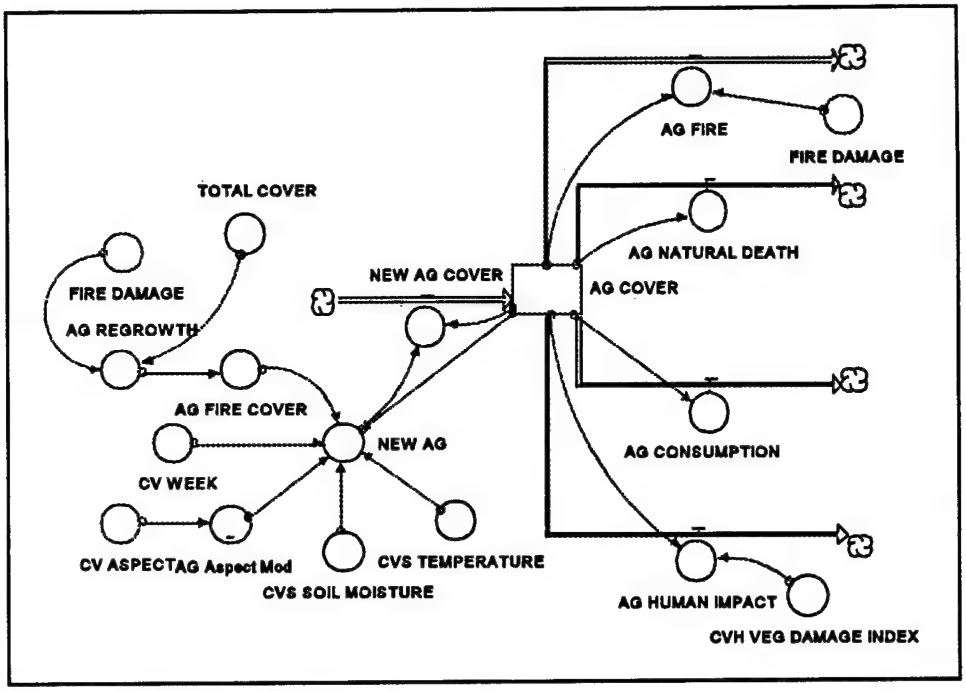


Figure 9. The grass submodel.

During the growing months, water is usually not limiting for grass growth in this type of ecosystem (Miller 1986). Therefore, the soil moisture variable was set for NEW_AG at zero. Also, the average optimal temperature for growth is between 20 and 30 °C, and growth occurs during the months of May through mid-July (Daubenmire 1972; Harris 1967). These grasses were thought to form a stable community where growth equaled death over the year, unless there was a perturbation such as a fire or human impact. Growth was approximated at 0.5 percent per year and the maximum cover for a cell was 59 percent (YTC report 1989). In case of a fire, all vegetation was consumed before the regrowth process began. According to Humphrey (1984), regrowth after a fire for grass species equals 30 percent of the viable vegetative cover the first year and 33 percent the second. The initial vegetative cover was taken from GIS maps derived from satellite imagery. For cover regrowth after a fire, the equation was:

AG_FIRE_COVER = DELAY(AG_ REGROWTH, 78) **[Eq 24]**

where AG_REGROWTH was defined as:

AG_REGROWTH = IF FIRE_DAMAGE=1THEN 0.33*TOTAL_COVER
ELSE 0 **[Eq 25]**

indicating that, after a burn, grasses return to about 30 to 33 percent of the initial cover (Humphrey 1984). NEW_AG and AG_COVER both fed into NEW_AG_COVER, which was defined as:

NEW_AG_COVER = MIN(NEW_ AG, AG_COVER - 59) **[Eq 26]**

indicating that total cover could never be more than 59 percent. The initial cover (AG_COVER) was determined through GIS and satellite imagery.

The only outflow different from the equation modeled for Sagebrush was natural death, defined as:

AG_NATURAL_DEATH = AG_COVER *.005/52 **[Eq 27]**

because, according to Treshow and Harper (1974), grass mortality is approximately 0.5 percent annually.

### *The Forbs Submodel*

The forbs model was similar in structure to the sagebrush submodel (Figure 10). Parameters that reflect growing conditions for the grass model were the same for the forbs model. Optimum growing conditions for forbs in this area were not represented

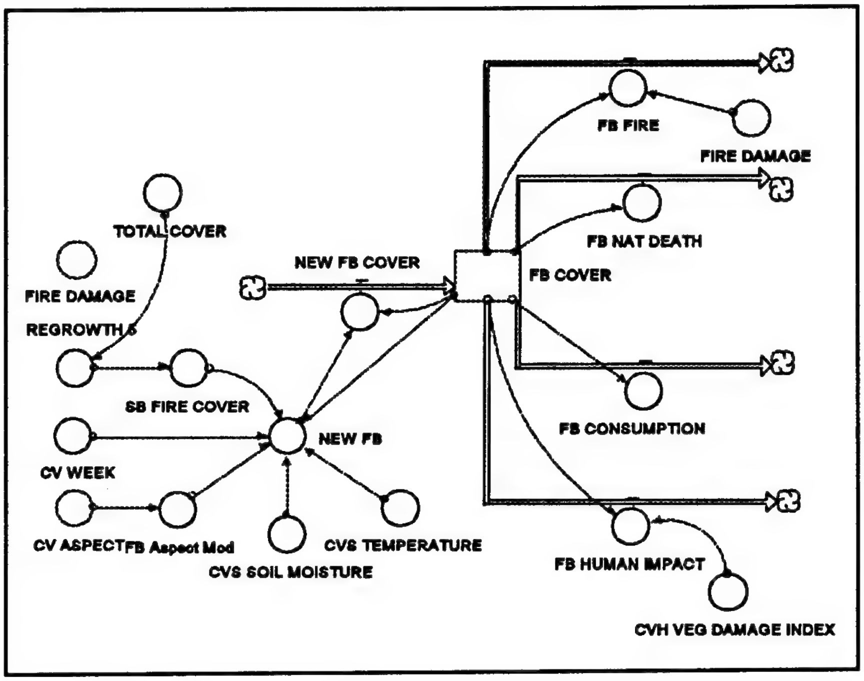


**Figure 10. The forbs submodel.**

in the data collected and therefore could not be incorporated. Data that signified optimum growing conditions for sagebrush were assumed to be indicative for all plant growth. (This specific data is available for inclusion into a future version of the model.)

The equation for new forbs (NEW_FB) was the same as that for sagebrush, with the same optimums set for time and conditions (temperature, moisture, and season). Some parameters that fed into NEW _FB were tailored specifically for the forbs model, e.g., RE_GROWTH_5, which was a part of the fire succession model and was defined as:

RE_GROWTH_5 = IF FIRE_DAMAGE = 1 THEN 0.6* TOTAL_COVER [Eq 28]
ELSE 0

reflecting the forbs ability to return to 60 percent of the initial cover after a burn (Humphrey 1984). The converter NEW _FB_COVER was similar since a maximum of 59 percent of the total cover could be forbs.

NEW_FB_COVER = MIN(NEW_FB, FB _COVER -59) [Eq 29]

The only outflow that changed in this model, as compared to the sagebrush model, was FB_NAT_DEATH. According to Treshow and Harper (1974), the mortality rate for forbs was approximately 10 percent each year due to natural death:

FB_NAT_DEATH = COVER 4*.1/52 [Eq 30]

The factor 0.1/52 should be changed if the DT is not set at 1 week. For example, if the DT is set at 1 day, then this factor should read 0.1/365.

## Physical Section

The central requirement of this model was to drive the weather affecting other submodels and specifically to drive the soil moisture cycle contained within the abiotic model. This physical process model reflected the measurable effects of the abiotic processes contained within the entire model.

Data from many sources were used to model this section. To include soil moisture, data on soil characteristics such as the maximum water holding capacity, the soil transmissivity (permeability), measurements of density or compactability, and the location of the various soil associations or series within the chosen study area were used. Spatial GIS maps provided soil characteristics such as slope, aspect, elevation, land use, and ground cover information.

GIS maps helped create a digital elevation model (DEM) that calculated change in precipitation and temperature over time. Temperature and precipitation data on average monthly values recorded in Yakima, WA were given in continuous graphs, and could be sampled over any time interval to estimate the amount of precipitation or average temperature (Mather 1965). The empirical relationships between these parameters, and how they changed given different slopes, elevations, and aspects were combined and implemented.

A wetness index of a given cell based on its slope and upslope area was also calculated from this data. This wetness index was used as a measure of water flowing through a given cell following rain. To model the relationship between vegetation and the soil moisture cycle, a cover factor was developed to reflect the relative amounts of grasses (*Agropyron*), forbs, and sagebrush present in a given cell.

Figure 11 shows an image taken from STELLA of the entire abiotic model. The model focused on the dynamics of AVAILABLE_SOIL_MOISTURE. Controlled principally by precipitation, temperature, and cover, this stock reflected the amount of available soil moisture (in millimeters of water) within the rooting zone and available for plant use. A depth of 15.0 cm represented the rooting zone. The model is described in terms of input and output. The input is precipitation (SM_ACTUAL_INFIL) and the output is runoff and evapotranspiration (SM_DECREASE). Each of these flows is discussed in the following sections.

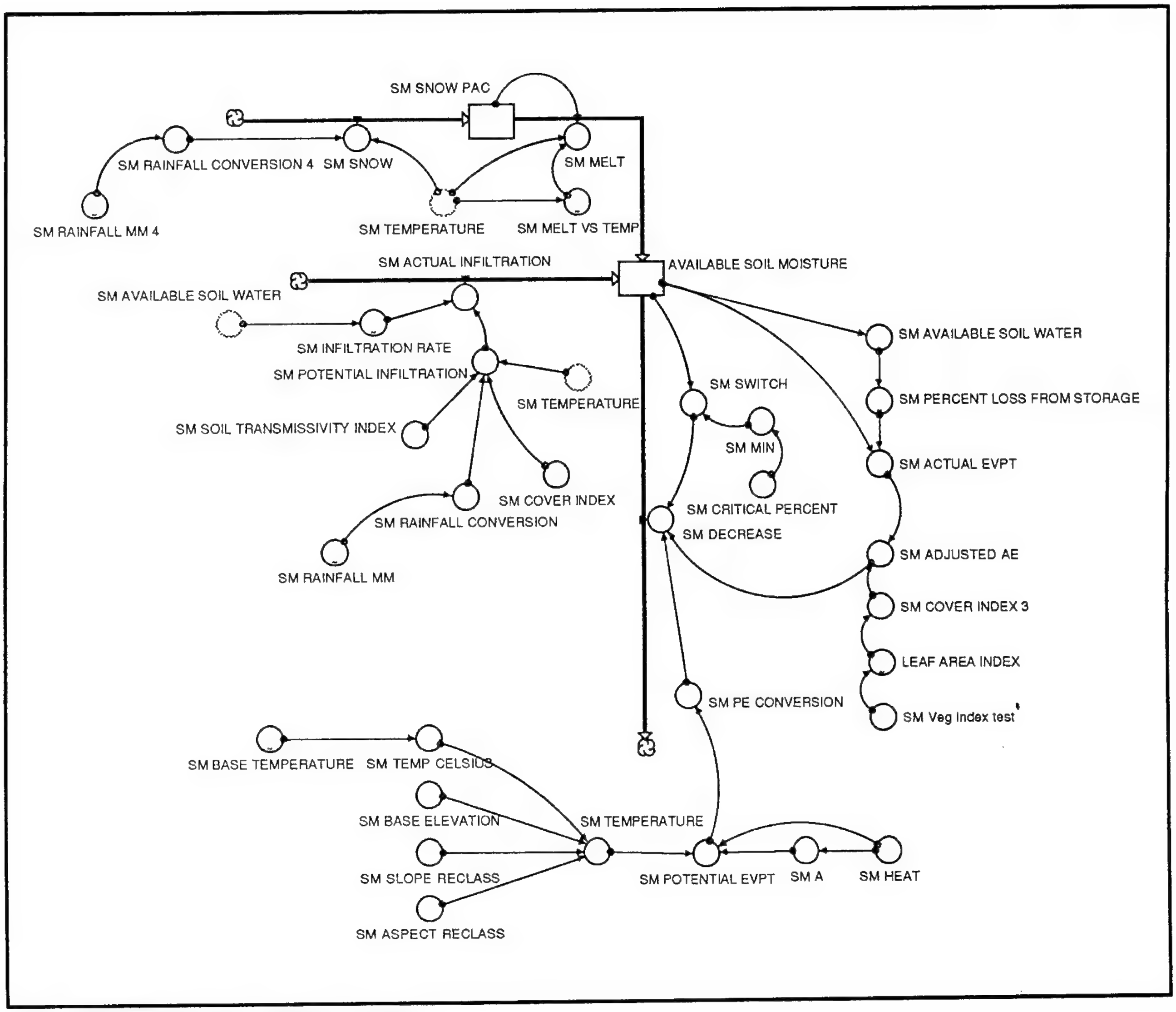


**Figure 11. Available soil moisture model**

### *Available Soil Moisture*

The main input and source of soil moisture increase came from precipitation and infiltration (Figure 12).

AVAILABLE_SOIL_MOISTURE was a function of SM_ACTUAL_INFILTRATION, which in turn was derived from the combination of SM_POTENTIAL_INFILTRATION and SM_INFILTRATION_RATE.

The calculations for the potential infiltration were derived from many factors, one of which was precipitation. Precipitation is the amount of water, liquid or frozen, that falls on the ground. Water supply, however, is the amount of liquid water that reaches

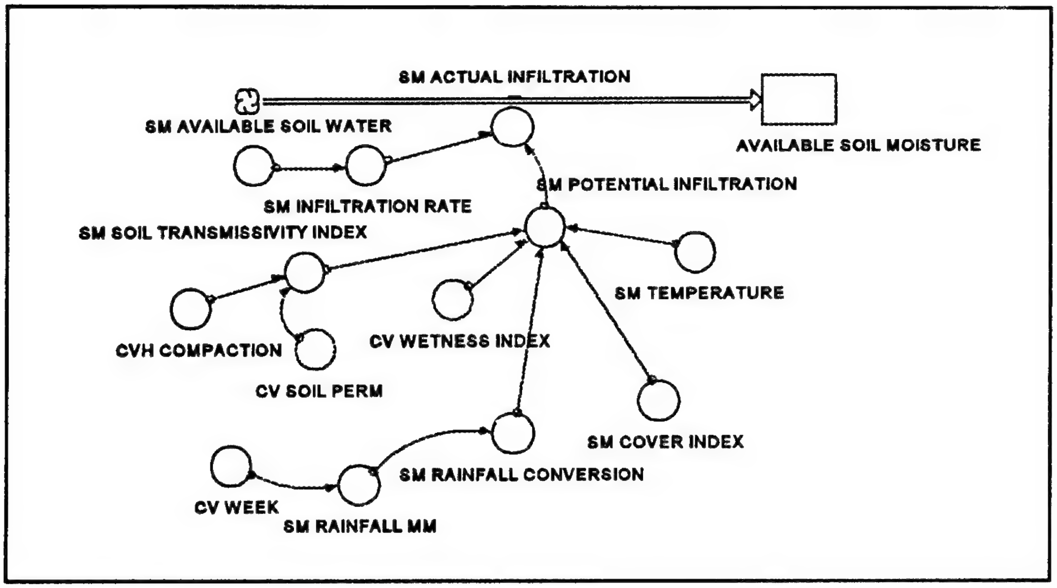


**Figure 12. Precipitation-driven input.**

the soil surface in a given period, consisting of both rainfall and meltwater. In this sense, snow is precipitation at the time it falls (unavailable to soil), but it is not considered a part of the water supply until it melts. At Yakima, the main water supply input comes from summer rains and the melting of the winter's snow. For this reason, the snow pack was another important input to the water supply and will be discussed later.

Data for the average monthly precipitation and temperature (SM_TEMPERATURE) was taken from Mather (1965). The original precipitation data was provided in SM_RAINFALL_MM (Figure 2), using a graph of average monthly precipitation vs time. Precipitation data was made usable in weekly time-steps with SM_RAIN-FALL_ CONVERSION:

SM RAINFALL CONVERSION= {SM_RAINFALL_MM/ 4.3}
+ {RANDOM (-0.65, 0.65)} * 0. [Eq 31]

The average monthly precipitation values were divided by 4.3 (the average number of weeks in a month) to permit calculating precipitation levels on a weekly rather than monthly basis. In addition, the equation allowed for the calculation of random variation. This process could be shut off by multiplying by zero (shown in the equation) or turned on by multiplying by 1.

The amount of water that actually soaked into the ground was not equal to the total amount of precipitation. First, topography affects water flow, a factor incorporated through the use of the wetness index calculated by GRASS, labeled CV_WETNESS_ INDEX in the model. The wetness index was a value (0.0 to 1.0 after normalization) that represented the actual amount of water flowing through a given cell following rain, and was based on the slope and location of the cell. A high wetness index meant more water was flowing through a cell during rain.

To completely calculate a value for SM_POTENTIAL_INFILTRATION, any effects of human activity (e.g., troop movements, tank travel, etc.) on the water-holding capacity of the soil must also be included. In this case, CVH_COMPACTION from "Human Impact" (p 42) was used to reduce the soil's transmissivity, labeled as SM_SOIL_TRANSMISSIVITY_INDEX. Both the SOIL_COMPACT_INDEX and the SM_TRANS_INDEX ranged from 0.0 to 1.0. The transmissivity was also affected by the soil permeability, CV_SOIL_PERM. Soil permeability must be derived from a GIS soils map. The equation that gave soil transmissivity was:

SM_SOIL_TRANSMISSIVITY_INDEX= (1-CVH_COMPACTION) [Eq 32]
* CV_SOIL_ PERM.

Soil transmissivity could be found by computing [ln (Te/Ti)] where Te was the average transmissivity and Ti was the transmissivity of the specific cell. This would be both spatially variable across cells, due to soil type and textural qualities, and temporally variable due to varied occurrences and intensities of human disturbance.

The final index input into potential infiltration was SM_COVER_INDEX, derived from the vegetative cover in the cell. This index allowed cover to be both spatially and temporally variable. An increase in cover lowered, and a decrease raised the infiltration.

When calculating the potential infiltration, researchers assumed that water would only infiltrate into the soil when the temperature was above freezing. This was included in the equation for potential infiltration, calculated as the product of the precipitation, wetness index, and transmissivity index when the temperature was greater than 0 °C:

SM_POTENTIAL_INFILTRATION = SM_TEMPERATURE [Eq 33]
SM_COVER_INDEX * CV_WETNESS_INDEX * SM_NEW_WATER
SM_RAINFALL_CONVERSION * SM_SOIL_TRANSMISSIVITY_ INDEX

The next section determined the infiltration rate, which was calculated using the available water in the soil (SM_AVAILABLE_SOIL_WATER). The current water level in the soil was represented as a percentage of the maximum water-holding capacity. This value was converted by a graph into the infiltration rate (SM_INFILTRATION_RATE). This function allowed for the rate of water infiltrating into the soil to decrease when the soil became more than half-full. The rate of infiltration continued to decrease rapidly until it stopped completely when the available soil moisture was equal to the maximum water-holding capacity of the soil. This causes the model to reflect the fact that soil more readily absorbs water when it is dry rather than wet.

The SM_ACTUAL_INFILTRATION was based on: (1) what might potentially infiltrate into the soil, and (2) the current level of water in the soil. The index for the potential infiltration rate and the infiltration rate were multiplied to give the actual infiltration:

SM ACTUAL INFILTRATION = SM_POTENTIAL_INFILTRATION * SM_INFILTRATION_RATE.SM_SNOW_PAC [Eq 34]

Figure 13 illustrates the processes of snow accumulation and snowmelt. The accumulation of snow in the model was recorded in the stock labeled SM_SNOW_PAC and was regulated by temperature and rainfall.

When precipitation occurs and the temperature is below 0 °C, snow starts to accumulate. If the temperature rises above 0 °C, then snow will melt and enter the soil as water. The amount of snow accumulation (SM_SNOW_PAC) was based on the standard that 1mm of rain equals 1mm of snow. Rainfall was taken from the precipitation data and the same conversions were used that calculated rainfall in mm/week (SM_RAINFALL_CONVERSION _4).

Like temperature and rainfall, the depth of snow was assumed to be constant over the entire area of a cell. The accumulation and depth of snow would be most strongly affected by the elevation of a cell because a decrease in temperature would accompany a rise in elevation. For this reason, the first freeze comes sooner and the last freeze is later at higher elevations.

When the temperature climbs above 0 °C, accumulated snow begins to melt, in an amount determined by the relationship described in the parameter SM_MELT_VS_TEMP as a graph. This graph showed that as the temperature climbed from 0 to 10 °C, the percentage of accumulated snow that became melt water increased exponentially from

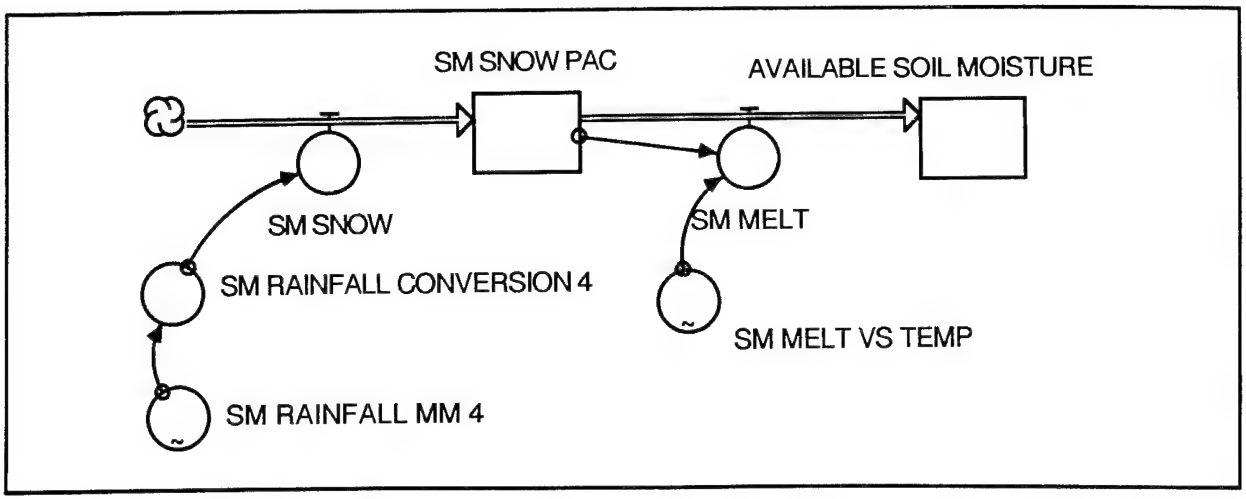


Figure 13. Snowfall and snow melt.

zero to 1. This means that no snow remained on the ground if the temperature climbed over 10 °C. The amount of water that melted was determined by the equation in SM_MELT:

```
SM_MELT = IF SM_TEMP > 0 THEN (SM_MELT_VS_TEMP                    [Eq 35]
 * SM_SNOW_PAC) ELSE 0
```

This equation gave the amount of meltwater from snow and transferred it to SM_NEW_WATER by:

```
SM_NEW_WATER= SM_RAINFALL_CONVERSION - SM_SNOW                    [Eq 36]
 + SM_MELT
```

This result was added to SM_POTENTIAL_INFILTRATION where it was incorporated into the total for soil moisture.

### Evapotranspiration

The rest of the model revolves around the theories of evapotranspiration and the water balance approach. Descriptions of the potential and actual evapotranspiration sections of the model, in addition to the switching mechanism between the two, will be explained in the discussion of the submodel.

The model that predicted available soil moisture was based on the dynamics of the water budget or climatic water balance. This water balance was defined as the interactions of energy and water described by the relationships among potential evapotranspiration, actual evapotranspiration, temperature, and precipitation. They were the most important parameters in the water balance, and also determined the availability of moisture in the soil. Water balance parameters estimated how much usable energy and water was available to plants, how much evaporative demand was not met by available water, and how much water was unusable excess. The idea of representing moisture availability as an energy/water index came from the early work of Thornthwaite (1948) and was further developed in Mather (1974, 1985) and Stephenson (1990). The next sections, which concern the model parameters, briefly explain the theory behind the equations.

### Soil Moisture

Soil moisture is the result of the balance between system input (water supply, i.e., precipitation and snowmelt) and output (potential and actual evapotranspiration), which vary both spatially and temporally, spatially because input and output are modified by terrain, temporally because monthly and weekly observations are sampled from annual cycles of precipitation and temperature (Figure 14).

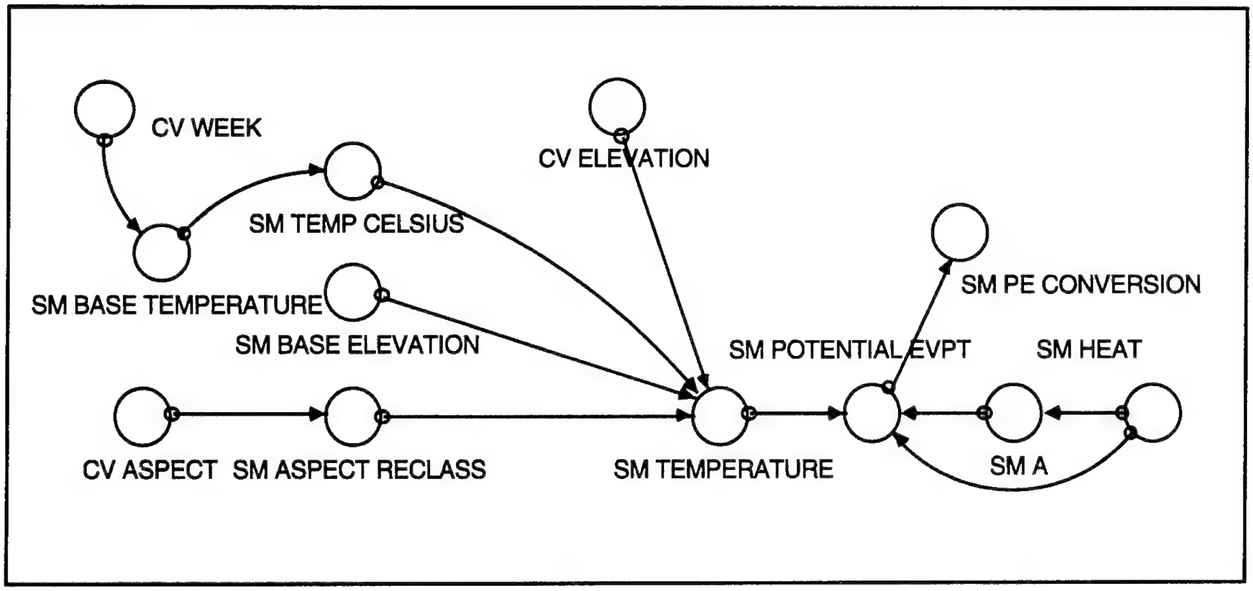


**Figure 14. Temperature.**

### *Potential Evapotranspiration*

Potential evapotranspiration (PE) is related to the amount of energy in the environment. Theoretically, PE is the evaporative water loss from a site with a standard vegetation cover supplied with unlimited water. Potential evapotranspiration is a function of heat (temperature) and radiation, but can be modified by air humidity and wind speed. For this model, PE was a direct function of air temperature only (a measure of heat) recorded as a monthly average. Its values were calculated according to the tables and methods described by Thornthwaite and Mather (1957). This method assumed that PE values could be obtained directly from air temperatures when the soil moisture retention occurred at a depth of 15 cm. Their units represented the variation in soil moisture in mm/month.

Air temperature of a cell was calculated based on its position in elevation relative to the base station at Yakima. Both the temperature and precipitation data were recorded at a weather station in Yakima at an elevation of approximately 365m above mean sea level and was assumed to have zero slope (no aspect). Temperature data were contained in the graph SM_BASE_TEMP and were recorded as a monthly average plotted over 52 weeks. This data was then converted from Fahrenheit to Celsius in SM_TEMP_CELSIUS.

The effects of elevation and aspect were combined in the SM_TEMPERATURE parameter according to the following relationships. For every 100m rise in elevation, the temperature fell 1 °C (for every 1m rise in elevation, the temperature falls 0.01 °C). The elevation was calculated by subtracting the base elevation (800m) from the actual elevation of the cell. This gave temperature differences from the base temperature value. Temperature was further modified based on the aspect of the cell. South facing slopes (225 to 315 degrees) were 10 percent warmer than the base temperature; north facing slopes (45 to 135 degrees) were 10 percent cooler; and east or west facing slopes (zero to 45, 135 to 225, or 315 to 360 degrees) were left unchanged. The final equation for temperature was:

```
SM TEMPERATURE = SM_TEMP_CELSIUS + ((CV_ELEVATION)                [Eq 37]
 * 0.01) * SM_ASPECT_RECLASS
```

In this model, the process of evapotranspiration was controlled principally by the air temperature. The actual amount of water that could potentially leave the system was calculated with an equation originally developed by Thornthwaite (1948). Aside from the temperature of the cell, the equation required two variables derived from the temperature, SM_A and SM_HEAT. Each of the following equations was taken directly from Thornthwaite and Mather (1955). The value used for SM_HEAT was calculated as:

```
SM_HEAT = (12/12) * ((118.44 / 5.0) ^1.514)                       [Eq 38]
```

and was given in degrees celsius. Twelve is the number of months over which balancing is to occur, and the 118.44 is the sum of the average monthly temperatures. This sum was assumed to be stable enough to use year to year. The SM_A was based on the SM_HEAT value:

```
SM_A = ((6.75/10.0^7.0) * SM_HEAT^3.00) - ((7.71/10.0^5.0) *      [Eq 39]
SM_HEAT^2.00) + ((1.79/10.0^2.0) * SM_HEAT) + 0.49
```

Using the values calculated by the SM_A and SM_HEAT parameters, the amount of potential evapotranspiration was determined with the following equation and measured in mm/month.:

```
SM_POTENTIAL EVPT = IF SM_TEMPERATURE > 0.00                      [Eq 40]
THEN 16.0 * ((8.0 * (SM_TEMPERATURE / SM_HEAT )) ^ SM_A) ELSE 0
```

The resulting value was converted to weekly values by dividing by 4.3 (for the same reason as was done for rainfall) in the SM_PE_CONVERSION parameter. The SM_PE_CONVERSION was the value used for potential evapotranspiration.

### *Actual Evapotranspiration*

Actual evapotranspiration (AE) is a measure of the simultaneous availability of both the biologically usable energy and water in the environment. This value can be obtained only when the water retention capacity of the soil is known. It equals the evaporative water loss from a site covered with a homogeneous vegetation (given water availability). Actual evapotranspiration equals either potential evapotranspiration or some fraction thereof determined by the relative energy demands of the system. For example, when PE exceeds available water, AE is limited by water and equals adjusted actual evapotranspiration. When available water exceeds PE, AE is limited by energy and AE equals PE.

The calculation of actual evapotranspiration was a two-stage process that began by modifying the potential evapotranspiration based on the relative amount of water available (SM_AVAILABLE_SOIL_WATER) given as a percentage of the maximum water-holding capacity of the soil or CV_WATER_HOLDING_CAPACITY and the AVAILABLE_SOIL_MOISTURE:

SM_AVAILABLE_SOIL_WATER= (AVAILABLE_SOIL_MOISTURE /CV_WATER_HOLDING_CAPACITY) * 100 percent **[Eq 41]**

This gave the percentage of potential saturation. SM_AVAILABLE_SOIL_WATER tells how saturated the soil is, and, when saturation was less than 35 percent, the amount of the potential evapotranspiration removed from the system and decreased based on the level of atmospheric demand. The percentage of decrease was graphed in the parameter SM_PERCENT_LOSS_FROM_STORAGE, which was derived from the SPAW (Saxton and McGuinnes 1982) and CREAMS models of evapotranspiration (Ritchie 1972). The percent loss from storage was multiplied by the available soil moisture to give SM_ACTUAL_EVPT:

SM_ACTUAL_EVPT= SM_PERCENT_LOSS_FROM_STORAGE *AVAILABLE_SOIL_MOISTURE. **[Eq 42]**

The second stage of this process involved a modification of the AE based on the percent cover present in the cell. A vegetation index was calculated based on the relative amounts of each type of plant compared to their initial proportions.

The vegetation index was then converted into a Leaf Area Index (LAI), which measured the percent of leaf area covering the ground. For example, if the vegetation index rose from 20 to 30, the LAI rose proportionally from 0.5 to approximately 3.5. The LAI was incorporated in an equation taken from Hanson (1976):

SM_COVER_INDEX 3 = 0.55 * (( LEAF_AREA_INDEX ) ^ 0.5 ) **[Eq 43]**

The LAI peaked at 3.3 because the cover index reached the limit of 100 percent. The final value calculated for actual evapotranspiration was labeled SM_ADJUSTED AE and was the product of the actual evapotranspiration and the cover index:

SM ADJUSTED AE= SM_COVER_INDEX_3 * SM_ACTUAL_EVPT [Eq 44]

This relationship states that, as the amount of cover increased, the percentage of the potential evapotranspiration actually removed from the soil increased until SM_ADJUSTED_AE equaled the SM_ACTUAL_EVPT. The actual evapotranspiration was directed into the SM_DECREASE, which was an outflow of the available soil moisture.

The switch controlled the evapotranspiration when it was equal to the potential evapotranspiration, SM_PE_CONVERSION, or the actual evapotranspiration, SM_ADJUSTED_AE (Figure 15). To create the switch, the SM_MIN, or minimum moisture content had to be determined. This was derived from the SM_CRITICAL_PERCENT and the CV_WATER_HOLDING_CAPACITY. The critical percent was the maximum storage capacity of the given soil type. If the moisture fell below this specified percentage, the plants were affected by the lower soil moisture. This stress created competition for the small amount of water still left in the soil. Therefore, the actual evapotranspiration no longer equaled the potential evapotranspiration. The equation that gave for minimum moisture was:

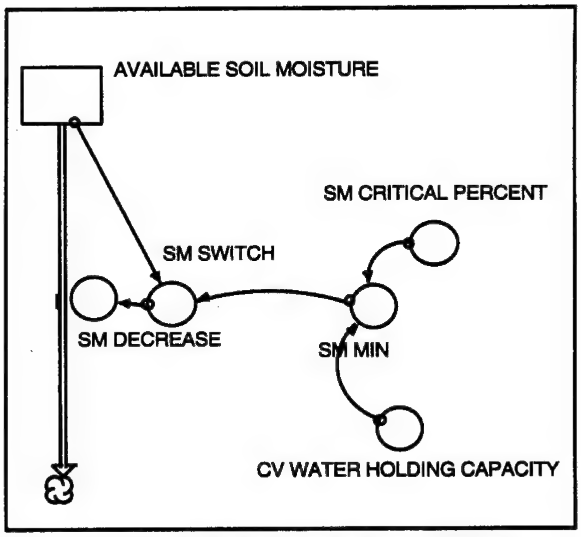


**Figure 15. The switching mechanism between potential and actual evapotranspiration.**

SM_MIN= SM_CRITICAL_PERCENT * CV_WATER_HOLDING_CAPACITY [Eq 45]

The SM_MIN was piped to the switch where it was compared with the available soil moisture.

SM_SWITCH= IF AVAILABLE_SOIL_MOISTURE < SM_MIN THEN 1 ELSE 0 [Eq 46]

When the switch equaled 1, factors comprising the actual evapotranspiration act decreased the available soil moisture in the cell. If the switch was zero, factors comprising the potential evapotranspiration decreased the available soil moisture. This relationship was given in the equation for SM_DECREASE:

```
SM_DECREASE = IF (SM_SWITCH = 1) THEN SM_ADJUSTED_AE                [Eq 47]
ELSE SM_PE_CONV.
```

which provided a value that decreased the available soil moisture of the cell.

This concludes the description of the actual modeling process and the components and relationships contained therein. As was previously explained, soil moisture was the result of the balance between system input (water supply) and system output (evapotranspiration). Spatial variations resulted from the varying input and output, which were modified by the terrain, and temporal variations were considered within monthly and weekly observations sampled from annual cycles of precipitation and temperature.

## Human Impact

An important aspect of this study was to model the potential effects of military training exercises on the Sage Grouse population, vegetation, and soil of the study site. Factors of training exercises that affect the environment included: (1) off-road vehicle use, (2) noise created by vehicles, (3) troop activities, and (4) encampments. Appendix B describes the GIS analysis steps used to generate impact maps that describe the result of a single type of training activity. The STELLA model components described here use these impact maps through a simple on-off mechanism. Note that only a single training scenario is thus available. Modifications to this model to improve realism will require the generation of a series of impact maps from different training activities. Three submodels describe these effects: soil compaction, vegetative disturbance, and noise effects. Each submodels accounts for environmental changes affecting the Sage Grouse.

Soil compaction affects soil moisture and productivity with regard to vegetative growth. Vegetative damage is measured because of its importance as food and cover for the Sage Grouse. Noise level affects the suitability of land for Sage Grouse habitation throughout the study site, particularly during the mating and nesting season. For each submodel section, a set of initial variables was used representing the effects of tracked and untracked vehicles, and troops and encampments on the site (Figure 16).

Data from the sources represented only extreme situations of military damage and did not contain intermediate results. When calculating damage per hour, data had to be extrapolated to include intermediate damage for a function to be graphed. The graph compared damage per hour with a coefficient between zero and 1. Units were created to define the relationships. A summary of the human impact parameters and their units are:

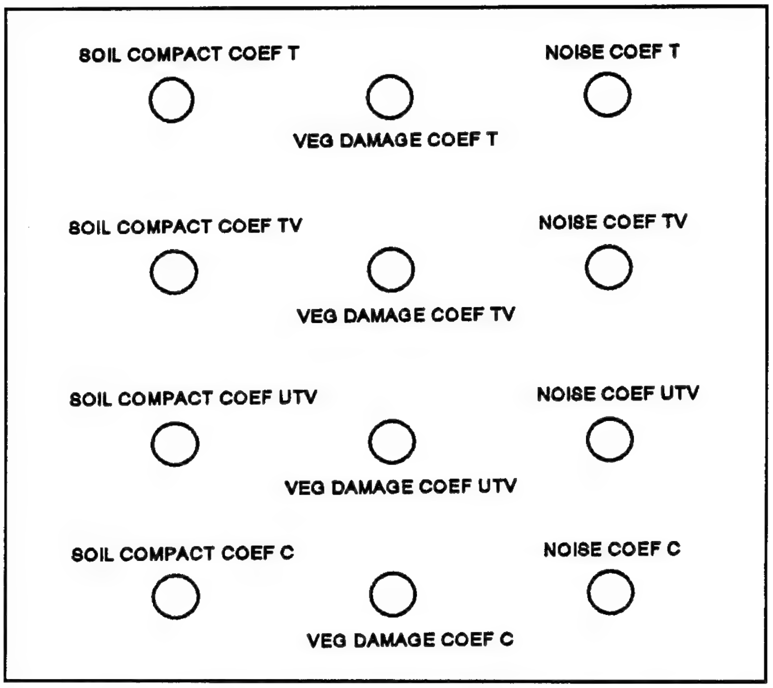


**Figure 16. Twelve initial parameters used for data in the human impact submodel.**

1. SOIL_COMPACT_COEF T = 0.0001: The soil compaction coefficient defined the compaction of soil caused by troops in hectares compacted per troop-hour of training.
2. SOIL_COMPACT_COEF_TV = 0.15: The soil compaction coefficient for tracked vehicles gave the hectares compacted per tracked vehicle hour of training.
3. SOIL_COMPACT_COEF_UTV = 0.10: The soil compaction coefficient for untracked vehicles gave the hectares compacted per untracked vehicle hour of training.
4. SOIL_COMPACT_COEF_C = 0.001: The soil compaction coefficient for encampment gave the compaction of soil due to the encampment of troops in hectares compacted per hour of encampment.
5. VEG_DAMAGE_COEF_T = 1: The vegetative damage coefficient for troops defined the vegetation destroyed per hectare per person per hour of training.
6. VEG_DAMAGE_COEF_TV = 1000: The vegetative damage coefficient for tracked vehicles gave the relative amount of vegetation lost per hectare per hour of vehicle use. By definition, tracked vehicles have 1000 times more impact on vegetation than troops.
7. VEG_DAMAGE_COEF_UTV = 500: The vegetative damage coefficient for untracked vehicles measured the relative amount of vegetation destroyed per hectare per hour of vehicle use. This estimate means that untracked vehicles were estimated to have 500 times more impact on vegetation than troops.

8. VEG_DAMAGE_COEF_C = 1: The vegetation damage coefficient was created from encampments. By definition, for every troop in an encampment there was one unit of vegetation destroyed per hectare per person per hour of training.
9. NOISE_COEF_T = 1: The noise coefficient for damage caused by troops is defined as noise units per troop hour of training. The units are termed troop-hour noise units.
10. NOISE_COEF_TV = 100: The noise coefficient for damage caused by tracked vehicles was defined in troop-hour noise units. It was estimated that tracked vehicles cause 100 times the noise output of a soldier.
11. NOISE_COEF_UTV = 50: The noise coefficient for damage caused by untracked vehicles was defined in troop-hour noise units. It was estimated that untracked vehicles cause 50 times the noise output of a soldier.
12. NOISE_COEF_C = 0.5: The noise coefficient for damage caused by a single soldier during encampments was defined in troop-hour noise units. It was estimated that encampments cause 0.5 times the noise output of a soldier.

This information was specific to the site under study, and parameters for training exercises and damage will have to be determined. Sources that provided data for these coefficients include: Bailey and Burt (1988), Bailey et al. (1988), Grassman et al. (1989), Griggs and Walsh (1981), Johnson and Burt (1990), Pollack et al. (1986), and Smith and Dickson (1990).

### *Training Submodel*

A training schedule was used as the input designed to trigger the running of the human impact model. The trigger indicated whether a training exercise was in progress at a given time. A time series graph was constructed to define how often an exercise took place in the model such that an output with a value of 1 signified a full training exercise and a zero signified no training. A training exercise was comprised of troops accompanied by tracked and untracked vehicles maneuvering through a landscape off and on-road. The training schedule was transformed into the number of tracked and untracked vehicles hours, the number of troops, and the number of troop encampment hours per 100 hectares per cell (as convenient and accessible units) in the cellular model. The noise intensity related to these activities was also provided for each cell.

Soldiers must train at various times during the year under a variety of climatic conditions. These exercises affect the desert ecosystem through soil compaction, vegetative damage, and noise, which can all be minimized by manipulating the timing of the training. The spatial distribution of a training exercise was determined outside the model and was fixed. Updates to this model should accommodate different spatial training locations and types of training activities.

### *Impact Submodels*

Through the use of the initial variables and other factors, three submodels were created to evaluate compaction, vegetative loss, and noise impact. Figure 17 shows the soil compaction submodel.

The soil compaction submodel (indicated with "SOIL_COM") was divided into four sections that created multipliers (indicated by "MULT"): SOIL_COM_C_MULT (encampments), SOIL_COM_T_MULT (troops), SOIL_COM_TV_MULT (tracked vehicles), and SOIL_COM_UTV_MULT (untracked vehicles). Each multiplier had two factors: the soil compaction coefficient of the impact type, and the corresponding number of training hours. Since the logic is identical for all of the multipliers, only the SOIL_COM_TV_MULT is discussed in detail.

SOIL_COM_TV_MULT was used to determine the level of tracked vehicle activity occurring in a cell. This multiplier captured the activity level by multiplying the coefficient, which represented the potential damage of 1 vehicle-hour, by the number of tracked vehicle-hours given in hours of vehicle use per 100 hectares. The result was a number that represented the complete effect of tracked vehicle use on the site. The following equation calculated the multiplier and gave the units in hectare/hours:

SOIL_COM_TV_MULT = SOIL_COMPACT_COEF_TV * CV_TRAK_VEH_HR/100 [Eq 48]

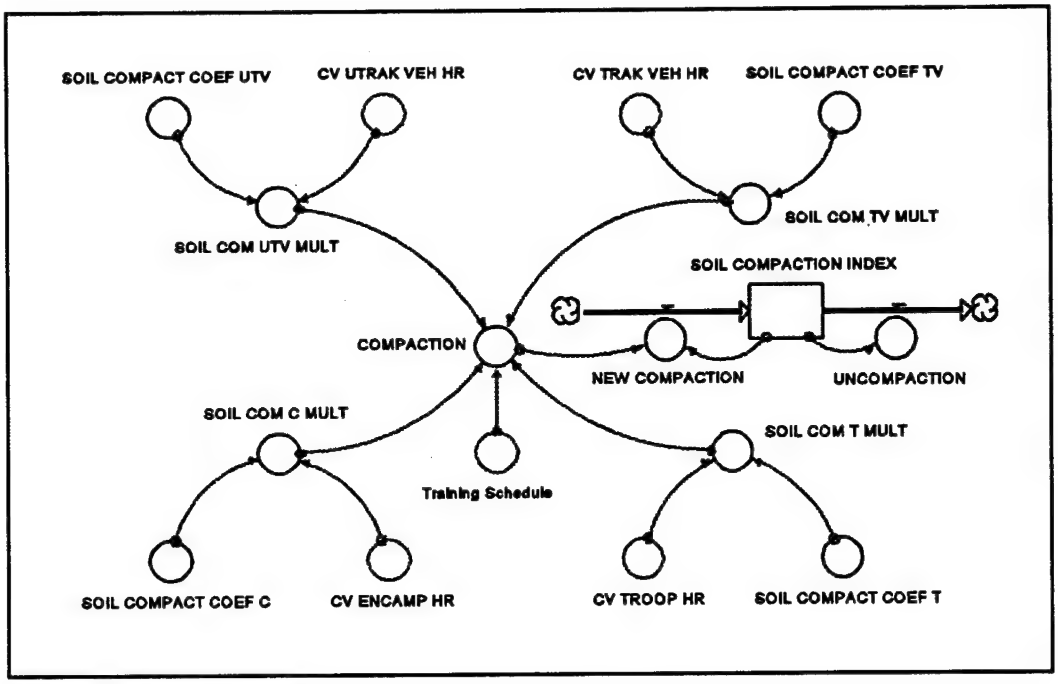


Figure 17. Soil compaction.

Because the logic was exactly the same for the other multipliers, their formulas in STELLA were very similar. The only difference was that each multiplier corresponded to its own soil compact coefficient value and number of training hours. The formulas for the remaining three multipliers were:

SOIL_COM_C_MULT = CV_ENCAMP_HR * SOIL_COMPACT_COEF_C/100 [Eq 49]

SOIL_COM_T_MULT = SOIL_COMPACT_COEF_T* CV_TROOP_HR/100 [Eq 50]

SOIL_COM_UTV_MULT = SOIL_COMPACT_COEF_UTV * CV_UTRAK_VEH_HR/100. [Eq 51]

The four multipliers were used as input to calculate the total compaction value (COMPACTION in Figure 17). COMPACTION passed the sum of its input values through an internal graph that yielded a value between zero and 1 (y-axis). The x-axis of the graph was described as:

Training_Schedule*(SOIL_COM_TV_MULT + SOIL_COM_UTV_MULT+ SOIL_COM_C_MULT + SOIL_COM_T_MULT)/1000 [Eq 52]

This graph is shown in Figure 18.

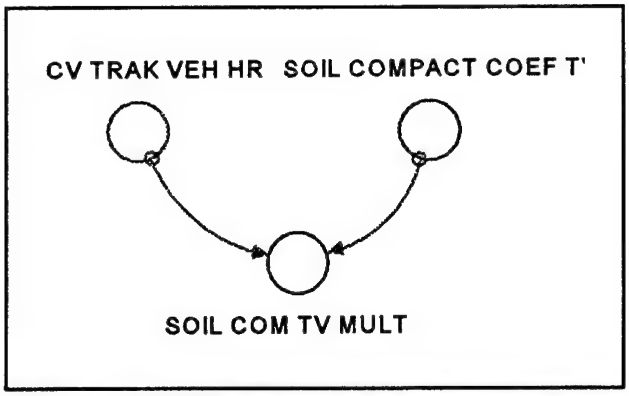


Figure 18. Soil compaction multiplier for tracked vehicles.

The construction of the NOISE_ INDEX and VEG_DAMAGE models was the same as the COMPACTION model up to this point, except that they used their own corresponding inputs for each model. (For example, NOISE_INDEX depended on the noise multipliers and noise coefficients). Figure 19 shows further modifications to the soil compaction model.

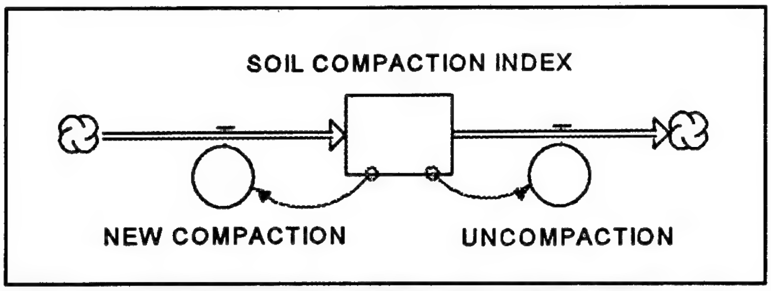


Figure 19. The soil compaction index portion of the model.

The value of COMPACTION represented the percentage of uncompacted land that was compacted as a result of a training event. Once the compaction index was calculated, its output was directed into NEW_COMPACTION. This computed the amount of currently uncompacted land that would then be compacted:

$$\text{NEW\_COMPACTION} = \text{COMPACTION} * \\ (1 - \text{SOIL\_COMPACTION\_INDEX}) \qquad \text{[Eq 53]}$$

The resulting value was entered into the SOIL_COMPACT_INDEX. The SOIL_COMPACT_INDEX represented the percentage of land compacted. The index generated UNCOMPACTION, the amount of land uncompacted every DT. This was calculated by subtracting a percentage of the index:

$$\text{UNCOMPACTION} = \text{SOIL\_COMPACT\_INDEX} * 0.01 \qquad \text{[Eq 54]}$$

# 4 The Migration Algorithm

To bring the cellular Grouse model to life, it was necessary to design an algorithm that could use the information generated within STELLA using the STELLA II software for the Macintosh. In addition, it was important that these algorithms could be implemented on either a serial or parallel processing platform. For this study, the CM–5 served as the parallel environment. The code was written in FORTRAN, with each section having a parallel and serial description of the algorithm.

The main function of the program was to calculate the movement of the female Sage Grouse from cell to cell. The assumption was made that, when mating season was in effect, there would be enough male Sage Grouse to mate with the females that arrived on the lek. The model allowed for movement in only the four cardinal directions. The algorithm, however, was constructed to allow for future adaptations.

The movement of a Grouse was influenced by the relative attractiveness of the neighboring cells versus the current cell. Movement varies based on Grouse population, noise levels, environmental factors (i.e., snow and sage brush cover), and proximity to the lek during the mating season. Furthermore, several rules were applied to affect how the Grouse could and would move. Both a "general_attraction" and a "lek_attraction" index were calculated within STELLA for use in the algorithm. During most of the year, the attractiveness of a neighboring cell and the current cell was based on the general_attraction index. However, during the mating season, from weeks 4 to 16 in the STELLA model, the level of the lek_attraction index determined the attractiveness of a cell. When lek_attraction began to rise (week 4) the nonpregnant female Grouse responded solely to this index, but once the Grouse was pregnant, the operative index switched back to general_attraction.

A third index considered after the evaluation of the relative attractiveness of the neighboring cells was the Grouse's level of motivation or "desire_to_migrate" (as it was named in the STELLA model). The specific functioning of these indices are described in detail in the following sections. It was required that only one adult Grouse could occupy any cell at one time. This rule was temporary suspended on 1 January when all of the juveniles traveling with the mother change status from juvenile to adult. Then the new adults in the cell with the mother scattered in all four directions to reduce the number of adult Grouse in each cell to one as quickly as possible. Also, due to the one-

Grouse-per-cell rule, if two Grouse wanted to move into the same cell at the same time, neither was permitted to move into the cell, and both stayed in their current cells. Lastly, no Grouse were permitted to enter or leave the study area.

The migration algorithm functioned in a two-stage process. Two passes were made over the data array to generate a new map of where the Grouse had moved following each iteration. The first pass calculated the "intent" of the Grouse. Depending on the relative attractiveness of the neighboring cells and the level of motivation a Grouse had to leave its current cell, a new array was generated. Each cell in this new "whereto" array contained the following information: (1) how many Grouse wished to move into a cell, and (2) from which direction a Grouse wanted to move. The second pass was intended to actually "move" a Grouse from its current cell to the desired cell, all rules permitting.

## Pass 1

The first pass began by checking every cell in the study area. When the first cell of the study area was encountered, it was examined to determine how many Grouse were in the cell. If the cell contained more than one adult, then the algorithm entered a subroutine to scatter all Grouse, except one, as quickly as possible.

This subroutine first evaluated the cell to the north. If this cell was in the study area and was unoccupied, then the whereto array made a note in the north cell (in the form of a unique number, i.e., north = 2) that one Grouse from the southern cell wanted to move into it. This same evaluation was carried out on the east, south, and west neighbors if there were still some Grouse to move. It was therefore possible to move four "new" adults out of a cell containing multiple Grouse in a single pass. If no Grouse were in the current cell, the algorithm moved on to the next cell and started again.

If, however, only one adult Grouse was in the current cell, the algorithm entered into a different subroutine. The first section of this subroutine assessed the motivation level of a Grouse to move from its current cell. The motivation index, as calculated in STELLA, was compared to a random number from 0.0 to 1.0. If the motivation level was less than or equal to the random number, then the Grouse did not move. Since the motivation index was also an index from 0.0 to 1.0, this random function proportionally decreased the probability that a Grouse would move as its motivation fell to zero. This introduced a level of variability into the movements of a Grouse. Even if there were a more desirable neighboring cell and a Grouse were sufficiently motivated to move, the randomness allowed for the chance that it would stay in its current cell.

Initially, the Grouse in the cell was checked to determine if it was already pregnant, and if it was currently the mating season. If this were the case, then the motivation assessment was bypassed, and the Grouse was guaranteed the option to move, one Grouse per cell, rule permitting. When this scenario was true, the attractiveness of the neighboring cells was calculated based on the general_attractiveness index. If however, the Grouse was either not pregnant or it was not the mating season, the motivation was compared to the random number to determine if the Grouse had a high enough motivation to move. If the Grouse were sufficiently motivated, the attractiveness of the current cell was calculated to compare it to the attractiveness of the neighboring cells.

If the Grouse were not pregnant and it was mating season, the lek_attractiveness index determined the attractiveness of the current and neighboring cells. If the Grouse were either pregnant or it was not mating season, the general_attractiveness index was used.

The next step in the algorithm determined which neighboring cell was most attractive compared to the current cell. Each of the north, south, east, and west cells was evaluated the same way. The method (lek_ versus general_ attractiveness) for calculating the attractiveness of the neighboring cell was the same as that for the current cell. An additional variation was incorporated by choosing a number for the neighbor's attractiveness based on a normal distribution. The distribution's mean equaled the difference in the attractiveness between the current cell and a specific neighbor cell. In addition, the curve had an adjustable variance. This effectively permitted the possibility that a Grouse would not move into the cell with the highest calculated attractiveness. Following the evaluation of all four neighboring cells, the cell with the highest chosen (rather than calculated) attractiveness ranking received a number in the whereto array signifying that, for example, a Grouse from the south wanted to move north into it. Lastly, if another Grouse wanted to move into that same cell, the value in the cell increased (by addition). In the second pass, the algorithm recognized that this number indicated two Grouse wanted to move into the same cell, and allowed neither to move.

## Pass 2

The second part of the algorithm was dedicated to the second pass over the data, and evaluated the newly generated "whereto" array. First, each cell of the whereto array was examined to determine if the cell: (1) was in the study area, and (2) contained a value that signified that some Grouse had chosen to move into the cell. Next, assuming the cell passed these two qualifications, its value was compared to those symbolizing the four cardinal directions. If the value did not exactly equal one of these, the cell was skipped and no Grouse were permitted to enter it. This occurred when the value in the cell was equal to the sum of two or more numbers which symbolized north, south, east, and west. However, when the value in the current cell of the whereto array did equal (e.g., north), a note was made in the current cell that a Grouse from the south (the current cell's southern neighbor) wanted to move north into the cell.

Taking this information, the southern cell was first checked to determine if there were more than one adult Grouse in the cell. Again, this would be the case only during the first few days of January when the juveniles had just become adults and all shared the cell with the mother. If there were more than one adult Grouse in the southern cell, then the number of adults in the current cell was set to 1 and the number of adults in the southern cell was reduced by one. Also, the pregnancy, juvenile, and egg variables that traveled with the adult Grouse as it moved throughout the year, were all set to zero following the move. For the majority of the year, however, this was not so.

More often than not, there would be only one adult Grouse in a neighboring cell. In this case, the Grouse, along with its associated pregnancy and juvenile values, were transferred from the neighbor to the current cell. These same values were then reset to zero in the neighboring cell from which it moved. The algorithm was also written so that, if a Grouse with eggs were for some reason (most likely human influence e.g., tank noise) scared into moving from her current cell, she would lose any eggs she was protecting.

Fear, or being "scared" is manifested in the STELLA model as the level of motivation for a Grouse to move. As the tranquility level (another STELLA index) of the current cell dropped, usually because of disturbance from Army training, the attractiveness of the current cell fell and the motivation to move from the cell increased. This reflected potential impacts of Army training or other human influences on the life cycle of the Sage Grouse. This process was then repeated from cell to cell of the whereto array until a new map had been generated showing the new locations of every Grouse following one run through the algorithm. The algorithm then returned to the beginning and ran again. Though the time-step or DT of the STELLA model was 1 week, the algorithm currently executes itself 21 times during that DT. The modeler can vary the number of loops.

Figure 20 shows a sample run of the above-described simulation. The nine frames represent the state of the simulation at different time steps. In these figures, the complete historical paths of each Grouse are represented by the growing trails.

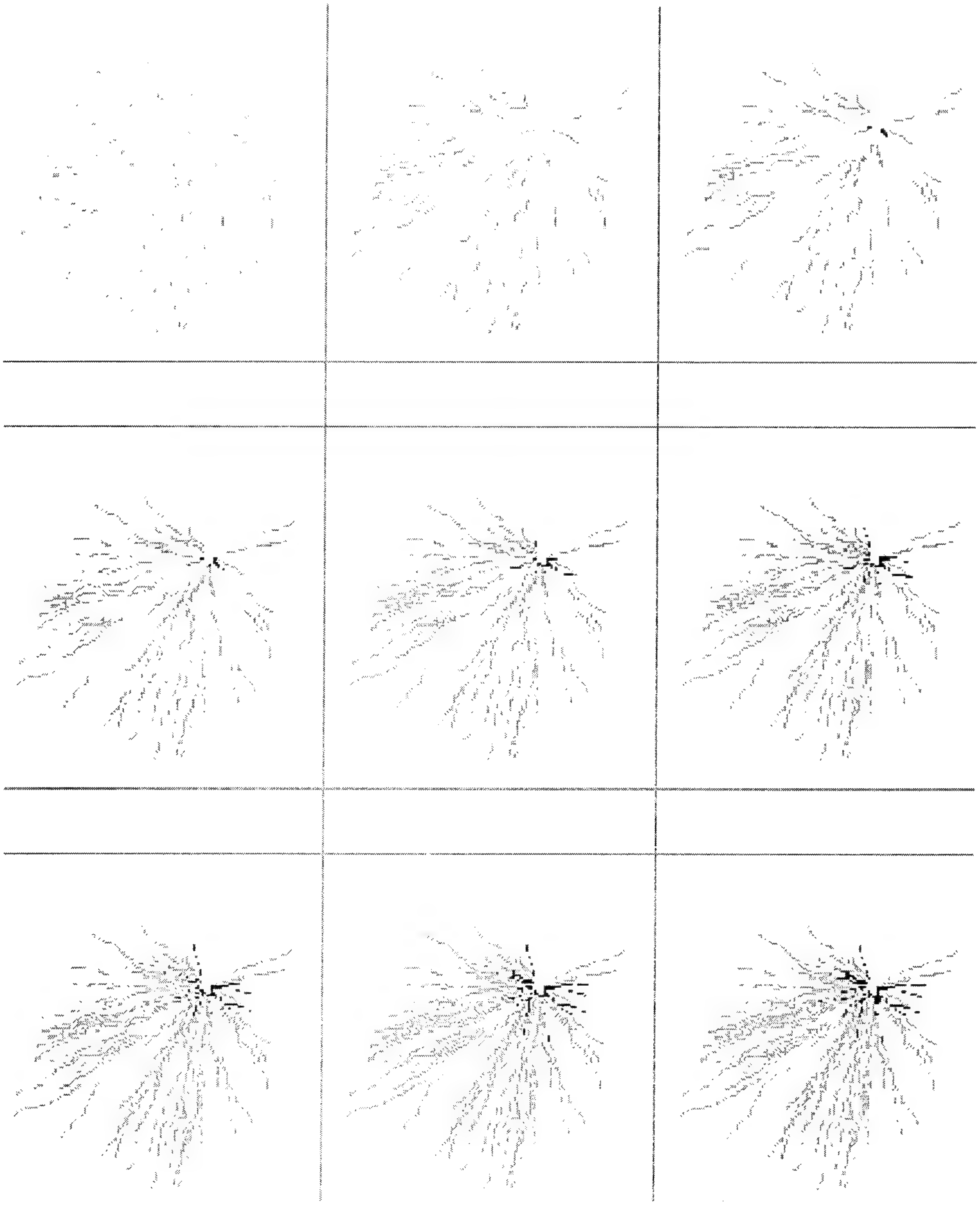


Figure 20. Grouse movement simulation: traces show grouse movement toward lek.

# 5 Working Model

The components described in the previous chapters define a model that estimates the human impacts of simulated training on a population of Sage Grouse through various cause-effect relationships. To simplify the model, these relationships were broken into two categories: long- and short-term impacts. The following long-term cause-effect chain of events was modeled by the researchers:

- Vehicle training destroys sage brush directly through trampling, thereby reducing the Grouse food supply, which in turn reduces the survival of Grouse.
- Similarly, the destruction of sagebrush increases the visibility of the Grouse to predators, thereby increasing death rates.
- Vehicle training compacts soil, altering its ability to absorb and retain rainwater. Less water is available to the sage brush and therefore affects available food and cover.
- Rainfall and snowmelt add water to the vegetation root zones. As the available moisture increases or decreases, the vegetation responds, thereby changing the food and cover potential.

The short-term cause-effect chain of events provided the following relationships:

- Vehicle training generates noise, which can frighten Grouse off nest sites, some sufficiently to abandon the nest. This leaves the eggs and chicks unprotected and they fall to predators or hunger.
- Snowfall hides the sagebrush food sources driving the Grouse to seek shelter and food elsewhere.
- The Sage Grouse migration pressures are a function of time of year, gestative state, and surrounding food and shelter potentials.

Debugging such a large model as the one described in this document is a tedious and expensive process, usually accomplished by reducing a large system to manageable pieces, which are then slowly rebuilt as subsections of the model are debugged and verified. For this study, the long-term cause-effect chains were deleted to yield a more streamlined short-term impact model that was easier to debug. The reduced model, which was successfully run as the final step of this project, is represented in Figure 20 and can be compared with initial model shown in Figure 1.

This STELLA model became the cellular model, which was applied simultaneously to each grid cell. The migration algorithm as captured in Fortran code was combined into a single computer-executable program with the STELLA model, which was also translated to Fortran. The result was then debugged and run on the CM-5 Connection Machine.

Several runs were generated with the completed model, three of which are reported here. The development of the model was based on the belief that nesting time is a critical time for the Grouse. Accordingly, the following test scenarios were designed:

1. No training
2. Training all year
3. Training all year except during the critical nesting season.

All system runs were conducted with 1-week time intervals covering an entire year beginning 1 January. Two hundred female Grouse were randomly distributed across the landscape to begin each simulation run.

Figures 21 and 22 show the results of each run; each figure contains an image of the system shortly after the nesting season and at the end of a full year of simulation. Note the visual increase in the number of Grouse for the simulations of weeks 48 and zero. At the end of each of these runs there exist clusters of Grouse representing a mother and a set of new adults that "fledge" from the mother at the changing of the calendar year. Also note that the Grouse are more scattered at the end of the training scenarios for weeks 48 and 52 as a result of a continued motivation to move around in response to training exercises.

Graphical results of these simulations are available through visualizations accessible on the INTERNET. Commonly available Mosaic programs allow interested readers to access the following Universal Resource Locator (URL): http://ice.gis.uiuc.edu

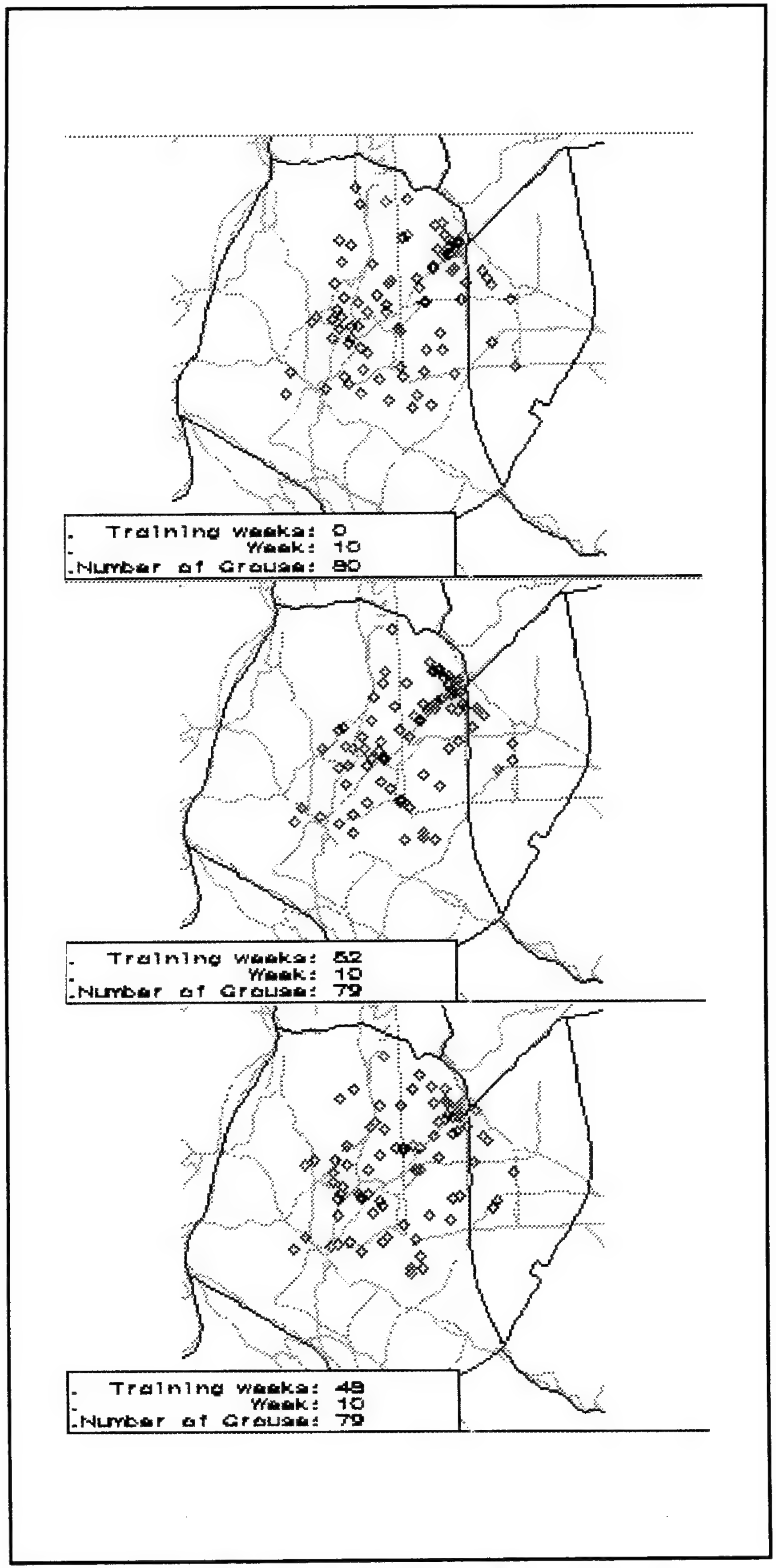


**Figure 21. System states after 10 weeks of simulation.**

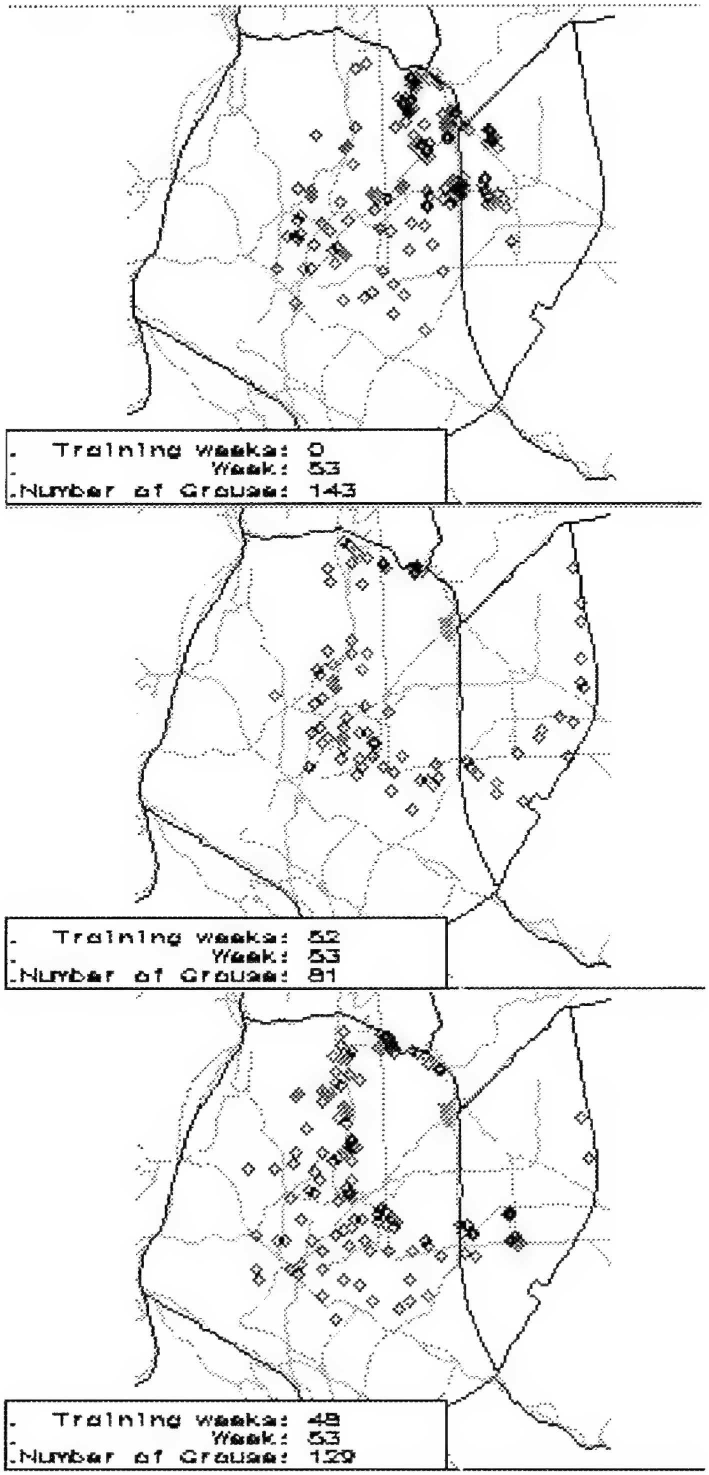


Figure 22. System states after 53 weeks of simulation.

# 6 Summary and Recommendations

## Summary

Traditional ecological models have focused on modeling the ecosystem as a homogeneous whole. This study demonstrated the feasibility of the cellular approach to the process of ecosystem modeling over broad spatial areas. This was accomplished by creating a working, dynamic spatial model of the ecosystem of the Sage Grouse at Yakima Training Center, WA. The important components needed to create the models demonstrated in this exercise were:

- *Modeling personnel*—Developing an ecological model is a multidisciplinary exercise requiring the coordination and cooperation of specialists from such diverse fields as ecology, biology, ethology, chemistry, agronomy, economics, landscape architecture, geography, geology, urban and regional planning, and civil engineering. The environment in which models are developed must be appropriate for multidisciplinary collaborative efforts.
- *Modeling process*—A modeling process must be established and well understood by the all modeling participants to allow full collaboration. This process must allow the model to be broken into components, often hierarchically, to allow individuals or small modeling subgroups to focus on a part of the large model in a way that allows an easy re-construction of components after development. This demonstration used a "base model" that established specific system output without any modeling. Sub-models were then developed within exact copies of this test (or template) environment. When it was time to plug the sub-models together, the integration was accomplished with minimal difficulties.
- *Modeling software*—The software environment makes it possible to collaborate effectively. For this exercise, that software environment was the STELLA software package. All participants had access to this software and were able to learn the software relatively quickly. People who had never programmed a computer were able to participate fully in the process of model design and development.

In the course of this project, the following software capabilities were collected into one research effort:

- *GRASS*—The Geographic Resources Analysis Support System was well suited to the development of the initial system state.
- *STELLA*—This graphical user-interface-oriented dynamic programming language enabled nonprogrammers with varying expertise to access the simulation models directly.
- *STELLA translator*—The STELLA translator as developed by Dr. Thomas Maxwell of the University of Maryland's International Institute of Ecological Economics converted the STELLA models into a form that could merged with other simulation code and run in parallel processing environments. A separate program to translate STELLA to Fortran was written for this exercise because the CM-5 had only Fortran compilers available.
- *Express software*—A more recent version of the STELLA translator made it possible to generate and run a STELLA-based cellular model on a variety of parallel processing hardware environments, including a network of UNIX workstations, connection machines, and networks of transputers attached to Macintosh or UNIX environments. This was made possible with a commercial package called Express.
- *Migration software*—The migration algorithm was the only component of the model generated by a researcher who specialized in writing computer software. Extended versions of this software are now possible that may facilitate the migration of multiple ecosystem components.

Note that this software was developed hierarchically to allow multiple participants to codevelop the system. Such a process requires system debugging at every level of the hierarchy. For example, once the individual developers created working model components within a STELLA sector, that sector had to then be debugged. Similarly, the simultaneous operation of a number of sectors generated new system states that could require changes to the lowest level of system components. Then, running the model simultaneously on almost 116,000 disconnected cells would reveal a number of additional model difficulties. Finally, connecting the cells with the independently programmed migration algorithm subjected the model to even more complex system states.

## Recommendations

This effort has demonstrated the technical capabilities of the modeling process. It has also shown how a multidisciplinary group of researchers can be coordinated to use a carefully integrated hardware and software system to create a successful working model a dynamic ecosystem. Further development of all aspects of the modeling effort remains.

If this technology is to have any potential as a management tool, it is recommended that the customer be the driving force behind model design and development. Like any solution offered to an end user, software can attain its useful potential only when the user provides the project's goals and participates in ensuring a satisfactory solution. For example, this demonstration's modeling of training activity presumed a single type of training exercise uniformly distributed across the landscape (a training and impact area) with probabilities based on a fixed tradeoff between travel time and training desirability. Installation personnel will be invaluable in the generating a series of much more realistic training activities and alternatives.

It is recommended that, if this model is to be practically applied to the Sage Grouse habitat at YTC, the following parameters be more thoroughly tested and verified:

- The Grouse behavioral model expressed in the migration algorithm—With significant input from the people that understand the details of Grouse behavior, the existing migration model should provide a good starting point.
- Growth parameters for the sage brush, agropyron, and forb growth submodels.
- Overland water flow during storm events for the precipitation and soil moisture models—With minimal parameterization, the precipitation and soil moisture models should perform quite satisfactorily.
- The impact of noise on the female Grouse, especially during the nesting and mothering phases—The primary impact modeled in the working model (which was a subset of the full model) relies on this parameter. Parameters should be developed to convert raw noise into Grouse annoyance levels either through field studies or through an interrogation of installation personnel familiar with Grouse behavior.
- How much acclimation to noise is possible, and how the noise translates to stress and survivability—Apparently the primary impact of noise on Grouse is to frighten Grouse away from their nests, which sometimes results in nest abandonment.
- Regrowth potential of sage brush after fire—since the model also considered fire.

While this effort has demonstrated powerful capabilities and techniques, improvements are recommended in the following technical approaches:

- Object-oriented design and development environment—Model components need to be encapsulated to allow for easier interdisciplinary team efforts. Submodels that did not communicate with other submodels under development were very easy to generate with the software used in this effort. Submodels, if turned into distinct objects, would force the modelers to more carefully identify inputs required from other model components and information about their components that is available to the rest of the model.

- Multiple dynamic time steps—Dynamic modeling, as used here, operated with a fixed time step. That is, the model operator must choose a fixed value (typically a week for this model) that represented the amount of time that passes between two time steps. Because (1) the activities occurring within the system may shift between fast and slow activities, and (2) the individual activities have innately different activity speeds, it becomes important to allow dynamically changing time steps as well as different internal time steps. The software in use for this exercise used a simple, predetermined, fixed-time step.
- Intercell interaction modeling—While STELLA was used to generate the cellular model, Fortran programming by a trained computer scientist was necessary to effect the generation of the intercellular model. This effectively removed the system modeler from the model and required close interaction and effort with the programmer. It is easy for the modeler to lose track of what the program is actually doing. In the future the modeler should have a more hands-on opportunity for direct ownership of the intercell models.
- Object libraries—All components of the models developed for this report were created "from scratch." While the final model is appropriate for this specific application, components of the model could be reused if developed within a modeling paradigm and language that stores and retrieves system components—involving object-oriented modeling with standalone objects that can be shared between similar ecological models.
- Probability/error computations—Models such as this derive their equations and rules from the results of experiments conducted within certain ranges of parameters. There is a fundamental error potential in these results based on statistical analysis of the experiments that provide the base data. Also, there is often an additional error associated with the proximity of the system state to the limits of the experimental conditions. It is inappropriate to extrapolate experimental results without recognizing an increasing error potential. Finally, as the inherent errors interact with each other in the model, the error of the final output should interest the modeler.
- Units management—The STELLA modeling environment provides a powerful environment for writing equations that describe the change in state from one time-step to the next. The software does not check the interaction of the units of measurements associated with the equations and hence leaves tremendous room for modeler error. Automatic checking and combination of errors would provide a beneficial capability to any such modeling environment.
- More rapid testing of the full model—Once the full model is assembled from its STELLA cellular models and Fortran encoded migration algorithms, the results need to be tested as a whole. Debugging results becomes tedious because the model is controlled by a computer scientist who must communicate the errors back to the modeling team. Potential fixes must be effected by the programmer and tested.

The turnaround time to the modeling team discourages efficient changes and modifications.

These modeling system suggestions will make the software environment easier and more efficient. The coming decade promises an explosion in the modeling of landscapes at all levels of resolution. This, in combination with more powerful, cost-effective computer hardware, will make dynamic, spatial, ecological modeling a key to better land management as land managers, in coordination with research institutions, develop more sophisticated and realistic models of local systems.

# Bibliography

Anders, F.J., and S.P. Leatherman, "Effects of Off-Road Vehicles on Coastal Foredunes at Fire Island, New York, USA," *Environmental Management*, vol 11, No. 1 (1987), pp 45-52.

Bailey, A.C., and E.C. Burt "Soil Stress States Under Various Tire Loadings," *Transactions of the ASAE*, vol 31, No. 3 (1988), pp 672-676.

Bailey, A.C., C.E., Johnson, and R.L. Schafer, "A Model for Agricultural Soil Compaction," *Journal of Agricultural Engineering Research*, vol 33, No. 4 (1986), pp 257-262.

Bailey, A.C., T.A. Nichols, and C.E. Johnson, "Soil Stress State Determination Under Wheel Loads," *Transactions of the ASAE*, vol 31, No. 5 (1988), pp 1309-1314.

Barker, J.R., and C.M. McKell, "Habitat Differences Between Basin and Wyoming Big Sagebrush in Contiguous Populations," *Journal of Range Management*, vol 36 (1983), pp 450-454.

Barker, J.R., and C.M. McKell, "Differences in Big Sagebrush (*Artemisia tridentata*) Plant Stature Along Soil-Water Gradients: Genetic Components," *Journal of Range Management*, vol 39 (1986), pp 147-151.

Beck, Thomas D.I., "Sage Grouse Flock Characteristics and Habitat Selection in Winter," *Journal of Wildlife Management*, vol 42, No. 1 (1977), pp 18-26.

Berry, J.D., and R.L. Eng, "Interseasonal Movements and Fidelity to Seasonal Use Areas by Female Sage Grouse," *Journal of Wildlife Management*, vol 49, No. 1 (1985), pp 237-240.

Biondini, M.E., C.D. Bonham, and E.F. Redente, "Secondary Successional Patterns in a Sagebrush (*Artemisia tridentata*) Community as They Relate to Soil Disturbance and Soil Biological Activity," *Vegetatio*, vol 60 (1985), pp 25-36.

Bonham, C.D., T.R. Cottrell, and J.E. Mitchell, "Inferences for Life History Strategies of *Artemisia tridentata* Subspecies," *Journal of Vegetation Science*, vol 2 (1991), pp 339-344.

Booth, G.D., B.L. Welch, and T.L.C. Jacobson "Seedling Growth Rate of 3 Subspecies of Big Sagebrush," *Journal of Range Management*, vol 43 (1990), pp 432-436.

Braun, C.E., T. Britt, and R.O. Wallestad, "Guidelines for the Maintenance of Sage Grouse Habitats," *Wildlife Society Bulletin*, vol 5, No. 3 (1977), pp 99-106.

Butler, C.D., et al., *A User's Guide for CAMMS Version 2.0* (Department of the Army: Army Corps of Engineers, 1989).

Cheney, R., *Memorandum for Secretaries of the Military Departments*, Subject: Environmental Management Policy (Washington, DC, 10 October 1989).

Connelly, J.W., W.L. Wakkinen, A.D. Apa, K.P. Reese, "Sage Grouse Use of Nest Sites in Southeastern Idaho," *Journal of Wildlife Management*, vol 55, No. 3 (1991), pp 521-524.

Connelly J.W., H.W. Browers, and R.J. Gates, "Seasonal Movements of Sage Grouse in Southeastern Idaho," *Journal of Wildlife Management*, vol 52, No. 1 (1988), pp 116-122.

Dalke, P.D., D.B. Pyrah, D.C. Stanton, J.E. Crawford, and E. F. Schlatterer, "Ecology, Productivity, and Management of Sage Grouse in Idaho," *Journal of Wildlife Management*, vol 27, No. 4 (1963), pp 811-840.

Daubenmire, R., "A Canopy-Coverage Method of Vegetational Analysis," *Northwest Science*, vol 33 (1959), pp 43-64.

———, *Steppe Vegetation of Washington*, Washington Agricultural Experiment Station Technical Bulletin 62.

———, "Annual Cycles of Soil Moisture and Temperature as Related to Grass Development in the Steppe of Eastern Washington," *Ecology*, vol 53 (1972), pp 419-424.

———, "Ecology of Artemisia *tridentata subsp. tridentata* in the State of Washington," *Northwest Science*, vol 49 (1975a), pp 24-35.

———, "Plant Succession on Abandoned Fields, and Fire Influences, in a Steppe Area in Southeastern Washington," *Northwest Science*, vol 49 (1975b), pp 36-48.

DePuit, E.J., and M.M. Caldwell, "Seasonal Pattern of Net Photosynthesis of *Artemisia tridentata*," *American Journal of Botany*, vol 60 (1973), pp 426-435.

Diersing, V.E., and W.D. Severinghaus, *The Effects of Tactical Vehicle Training on the Lands of Fort Carson, CO—An Ecological Assessment*, Technical Report (TR) N-85/03/ADA152142 (U.S. Army Construction Engineering Research Laboratory [USACERL], December 1984).

Donnelly, M., and J.G. Van Ness, "The Warrior and the Druid—DOD and Environmental Law," *Federal Bar News and Journal*, vol 33, No. 1 (1986), pp 37-43.

Donovan, L.A., and J.R. Ehlering, "Ecophysiological Differences Among Juvenile and Reproductive Plants of Several Woody Species," *Oecologia*, vol 86 (1991), pp 594-597.

Eberhardt, L.E., and L.A. Hoffman, *Sage Grouse on the Yakima Training Center: A Summary of the Studies Conducted During 1989 and 1990* (Battelle Memorial Institute, 1991).

Evans, R.D., R.A. Black, and S.O. Link, "Reproductive Growth During Drought in *Artemisia tridentata Nutt*," *Functional Ecology*, vol 5 (1991), pp 676-683.

Gassman, P.W., D.C. Erbach, and S.W. Melvin, "Analysis of Track and Wheel Soil Compaction," *Transactions of the ASAE*, vol 32, No. 2 (1989), pp 23-29

Goran, W.D., L.L. Radke, and W.D. Severinghaus, *An Overview of the Ecological Effects of Tracked Vehicles on Major U.S. Army Installations*, TR N-142/ADA126694 (USACERL, February 1983).

Griggs, G.B., and Walsh, B.L., "The Impact, Control and Mitigation of Off-road Vehicle Activity in Hungry Valley, California," *Environmental Geology*, vol 3, No. 4 (1981), pp 229-243.

Haferkamp, M.R., P.O. Currie, J.D. Volesky, and B.W. Knapp, "Mortality of Crested Wheatgrass and Russian Wildrye During Drought," *Journal of Range Management*, vol 45 (1992), pp 355-357.

Harris, G.A., "Some Competitive Relationships Between *Agropyron spicatum* and *Bromus tectorum*," *Ecological Monographs*, vol 37 (1967), pp 89-111.

Humphrey, L.D., "Patterns and Mechanisms of Plant Succession After Fire on *Artemisia*-Grass Sites in Southeastern Idaho," *Vegetation*, vol 57 (1984), pp 91-101.

Hupp, J.W., and C.E. Braun, "Topographic Distribution of Sage Grouse Foraging in Winter," *Journal of Wildlife Management*, vol 53, No. 3 (1989), pp 823-829.

Jenni, D.A., and J.E. Hartzler, "Attendence at a Sage Grouse Lek: Implications for Spring Censuses," *Journal of Wildlife Management*, vol 42, No. 1 (1978), pp 46-52.

Johnsgard, Paul A., *The Grouse of the World* (University of Nebraska Press, 1983), pp 112-126.

Johnson, G.D., and M.S. Boyce, "Feeding Trails With Insects in the Diet of Sage Grouse Chicks," *Journal of Wildlife Management*, vol 54, No. 1 (1990), pp 89-91.

Johnson, C.E., and Burt, E.C., "A Method of Predicting Soil Stress State Under Tires," *Transactions of the ASAE*, vol 33, No. 3 (1990), pp 713-717

Johnson, F.L., "Effects of Tank Training Activities on Botanical Features at Fort Hood, Texas," *Southwest Naturalist*, vol 27 (1982), pp 309-314.

Martin, N.S., "Sagebrush Control Related to Habitat and Sage Grouse Occurrence," *Journal of Wildlife Management*, vol 34 (1970), pp 313-320.

McArthur, E.D., and B.L. Welch, "Growth Rate Differences Among Big Sagebrush [*Artemisia tridentata*] Accessions and Subspecies," *Journal of Range Management*, vol 35 (1982), pp 396-401.

Miller, R.F., J.M. Seufert, and M.R. Haferkamp, *The Ecology and Mangement of Bluebunch Wheatgrass (Agopyron Spicatum): A Review*, Station Bulletin 669 (Agricultural Experiment Station, Oregon State University, Corvallis, 1986).

Peterson, J.G., "The Food Habits and Summer Distribution of Juvenile Sage Grouse in Central Montana," *Journal of Wildlife Management*, vol 34, No. 1 (1970), pp 147-155.

Pollock, D., J.V. Perumpral, and T. Kuppusamy, "Finite Element Analysis of Multipass Effects of Vehicles on Soil Compaction," *Transactions of the ASAE*, vol 29, No. 1 (1986), pp 45-50.

Remington, T.E., and C.E. Braun, "Sage Grouse Food Selection in Winter, North Park, Colorado," *Journal of Wildlife Management*, vol 49, No. 4 (1985), pp 1055-1061.

Sapsis, D.B., and J.B. Kauffman, "Fuel Consumption and Fire Behavior Associated With Prescribed Fires in Sagebrush Ecosystems," *Northwest Science*, vol 65 (1991), pp 173-179.

Schaeffer, D.J., et al., *Preliminary Study of Effects of Military Obscurant Smokes on Flora and Fauna During Field and Laboratory Exposures*, TR N-86/22/ADA174502 (USACERL, 1986).

Severinghaus, W.D., and W.D. Goran, *Effects of Tactical Vehicle Activity on the Mammals, Birds, and Vegetation at Fort Lewis, Washington*, TR N-116/ADA111201 (USACERL, November, 1981).

Severinghaus, W.D., R.E. Riggins, and W.D. Goran, *Effects of Tracked Vehicle Activity on Terrestrial Mammals, Birds and Vegetation of Fort Knox, KY*, TR N-77/ADA073782 (USACERL, July 1979).

Smith, D.L.O., and J.W. Dickson, "Contributions of Vehicle Weight and Ground Pressure to Soil Compaction," *Journal of Agricultural Engineering Research*, vol 46, No. 1 (1990), pp 13-29.

Treshow, M., and K. Harper, "Longevity of Perennial Forbs and Grasses," *Oikos*, vol 25 (1974), pp 93-96.

U.S. Army Corps of Engineers, *Draft Environmental Impact Statement Supplement-Fort Lewis Military Installation* (Fort Lewis and Yakima Firing Center, WA, 1986).

U.S.D.A. Forest Service, *Range Plant Handbook* (U.S. Government Printing Office, Washington, DC, 1937).

U.S. Department of the Army (DA), *Facilities Engineering and Housing Annual Summary of Operations, Fiscal Year 1989* (Office of the Assistant Chief of Engineers [OACE], USAEHSC, 1989).

Uresk, D.W., W.H. Rickard, and J.F. Cline, "Perennial Grasses and Their Response to a Wildfire in South-Central Washington," *Journal of Range Management*, vol 33 (1980), pp 111-114.

Wandera, J.L., J.H. Richards, and R.J. Mueller, "The Relationships Between Relative Growth Rate, Meristematic Potential, and Compensatory Growth of Semiarid-Land Shrubs," *Oecologia*, vol 90 (1992), pp 391-398.

Welch, B.L., and T.L.C. Jacobson, "Root Growth of *Artemisia tridentata*,“ *Journal of Range Management*, vol 41 (1988), pp 332-334.

West, N.E., K.H. Rea, and R.O. Harniss, "Plant Demographic Studies in Sagebrush-Grass Communities of Southeastern Idaho," *Ecology*, vol 60 (1979), pp 376-388.

Wright, H.A., and J.O. Klemmedson, "Effect of Fire on Bunchgrasses of the Sagebrush-Grass Region in Southern Idaho," *Ecology*, vol 46 (1965), pp 680-688.

Young, J.A., and R.A. Evans, "Population Dynamics After Wildfires in Sagebrush Grasslands," *Journal of Range Management*, vol 31 (1978), pp 283-289.

*Training Land*, Training Circular (TC) 25-1 (DA, Washington, DC, 1978).

*Yakima Training Center Range Site Description* Document No. 008BY010WA (1989).

# Appendix A: STELLA Software

The STELLA modeling software, operating in an Apple Macintosh environment, was chosen because this user-friendly combination of hardware and software facilitated the easy capture of the individual expertise of scientists who shared no common programming language skills and allowed them to cooperate in real-time modeling exercises. Such cooperation builds accurate, appropriate models in which all participants maintain joint ownership. Using simple graphical icons, modelers can rapidly generate the gross structure of a system.

Models were used to unfold a history of events based on the Euler and Runge-Kutta simulation techniques with four STELLA graphical components:

*Stock*—Represented by a rectangular box, the stock is a variable from which all computations in a time-step begin. A time-step represented the amount of time for the model to complete one cycle. Stocks are initialized at the start of a run with a value or an equation based the on starting values of other model components. A stock is represented in Figure A1 as POPULATION.

*Controller*—Controllers are "valves" that control the flow of values into and out of stocks and are driven by equations. BIRTHS and DEATHS in Figure A1 are controllers with arrows indicating allowable directions of flow.

*Converter*—Converters are similar to controllers but they do not directly control flow into or out of stocks. They are equations that result in the computation of

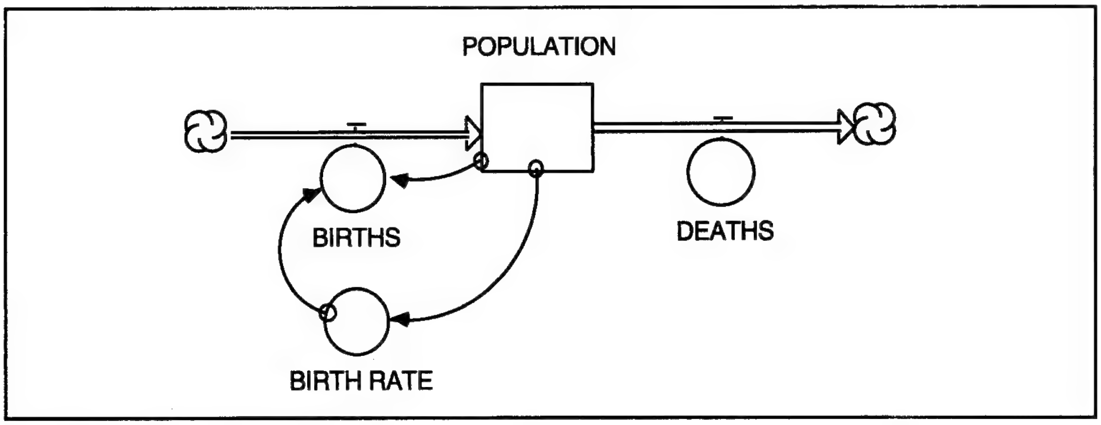


Figure A1. Simple STELLA model.

values that are themselves input to controllers or other converters. The example converter in Figure A1 is BIRTH_RATE, which is a function of POPULATION.

*Arrow*—Arrows connect stocks, controllers, and converters to controllers and converters. They provide a graphic to indicate that a controller or converter is a function of the collection of stocks, converters, and controllers that point to it.

Each converter and controller graphic can be enlarged to yield space for entering text to explain the item and to write the equation that is the function of the input. The equation options are flexible, allowing for basic arithmetic as well as a wide range of functions (statistical, business, trigonometric, probabilistic, and logic).

The STELLA modeling process and the Macintosh computer is too small to run a complex model simultaneously on hundreds of thousands of cells. Computational facilities, however, do exist in parallel processing computers, such as the CM-5. To use the CM-5, the programming equations generated in the STELLA modeling process were translated into another programming language with the goal to make the process as automatic as possible.

# Appendix B: Modeling Training

The following analysis steps were used to create the CV_TRAK_VEH_NOISE and CV_TRAK_VEH_HR maps:

The units generated for CV_TRAK_VEH_HR were total training seconds per cell for an exercise in which tracked vehicles run for 1000 hours getting to and from the training area and actually training (100 vehicles running for 10 hours).

Described first is the process for creating exercise time for vehicles. This script synthesizes a map of tank training hours per unit area (hectare). It is for demonstration purposes only; little connection to reality is claimed.

Assumptions:

1. Tanks will train closer to the cantonment rather than further preferentially on lower slopes.
2. One half of the total driving time takes place on the roads getting to and from the exercise area.

## Identify the Study Area

The analysis region for this exercise was defined as the smallest bounding box around 2A and 2B, which also contains a fork in the main road that accesses the areas. This fork is located at UTM coordinates 703573,5176033 and must be crossed by vehicles traveling between the training range and the installation cantonment areas.

A resolution of 30 m is chosen. Choosing a larger resolution will cause pieces of the roads to be lost, thus damaging the analysis. Smaller resolutions were judged to be unnecessary because of an increase the processing time with little extra advantage.

These requirements were captured with the GRASS command:

```
g.region n=5186260 s=5176000 e=709720 w=699460 nsres=30 ewres=30
```

Remove any MASK that might currently exist that would limit the spatial extent of the analysis.

```
g.remove MASK
```

## Generate Cost Surface Based on Travel Time

Because travel time to and from training exercises is a critical consideration based on a requirement to conserve time, fuel, and total tank driving hours, the first analysis requirement is to generate a travel-time map from a common point on the road to the cantonment area to all point within the training areas. To accomplish this, a map must be generated that identifies the travel time across each cell. Travel time for this analysis was simply a function of surface type for roads and slope for off-road travel. Defining a unit of speed as one cell per 2 seconds (30-meters/2-second or about 33 miles per hour), the following values were assigned to different land coverages:

1 unit of time to cross a cell on a surfaced road
2 units of time to cross a cell on a maintained road
5 units of time to cross a cell on a dirt road

These values needed to be applied to a map of roads which was created by changing a series of vector road maps into raster maps:

```
v.to.rast rds.dirt@grass out=rds.dirt
v.to.rast rds.maintain@grass out=rds.maintain
v.to.rast rds.surface@grass out=rds.surface
```

The travel time units were applied to these maps to create the map showing travel-time across each cell.

```
r.mapcalc << EOF
time.road = \
    if (rds.dirt,5, \
    if (rds.maintain,2, \
    if (rds.surface,1, 0)))
EOF
```

Cross-country travel times were simply reclassified slope values based on the following table:

10 units of time to cross a cell on slopes less than category 2 (5%)
20 units of time to cross a cell on slopes less than category 4 (15%)
40 units of time for higher slopes

These were applied using the following GRASS command:

```
r.mapcalc "time.slope = exp(slope@Grouse.model,2)"
```

Finally, the road travel times and the cross-country travel times were combined as follows. Note that areas outside the study area were simply assigned an arbitrarily high travel-time value of 1000.

```
r.mapcalc << EOF
time.combine = if (time.road, time.road, \
    if (time.slope,time.slope, \
    1000))
EOF
```

The final cost-surface map was then generated using the single-cell traversal time map and a starting point. The starting point was chosen as the point on the road closest to the cantonment area while still within the study area boundary. The GRASS command used for this step was:

```
r.cost time.combine out=time.cumulative coor=703573,5176033
```

## Generate Training Suitability Map

Training suitability is determined as a function of travel-time proximity, proximity to roads, and vegetative cover density. The first step is to force all further analysis to occur within the training area boundaries. This is accomplished by creating a map named MASK that has nonzero values for the training areas.

```
r.mapcalc << EOF
'MASK = if(tr.areas@PERMANENT == 4 || tr.areas@PERMANENT == 26,1)'
EOF
```

Resample cumulative map to delete areas outside training area:

```
r.mapcalc time.cumulative=time.cumulative
```

Find highest value in the cumulative time map to allow reversal of map values later in the analysis.

```
hi_time=`r.stats -q time.cumulative | tail -1`
```

Combine the travel-time map with the training suitability map. The coefficients were determined experimentally to generate a “proper” balance between the influence of slope and distance.

```
r.mapcalc suitable=“exp(30-slope@Grouse.model,3)/10 +
    $hi_time - time.cumulative”
```

Shift all values toward zero. This gives a bigger proportional advantage to those cells that are most suitable.

```
value=`r.stats -q suitable | head -1`
r.mapcalc “suitable = suitable - $value + 1"
```

## Generate Travel Time To/From Training

Assuming that the map just created is suitable and indicates a probability distribution of actual training, it is now possible to compute the time that will be spent traveling to and from these training areas. This is accomplished conceptually by placing “vehicles” in the training areas and allowing them to “travel” home via their respective least-cost paths. These paths follow a steepest descent route through the above generated “time.cumulative” map. This is done with the following commands:

```
r.mapcalc tmp=suitable/100.
r.watershed elev=time.combine flow=tmp accum=tank.accum
r.mapcalc “tank.accum=abs(tank.accum)”
```

This experiment worked with 100 vehicles. Hence, in travel to and from the training areas, the maximum number of tank “accumulating” will occur at the outlet (which will show the maximum traffic) and will be 200 vehicles. To get total vehicles visiting any given cell, adjust the maximum value to 200.

```
max_accum=`r.stats tank.accum | tail -1`
r.mapcalc “tank.en.route = (200. * tank.accum / $max_accum)”
```

## Generate Total Travel Time

Combine to/from travel times (tank.en.route) with training time (suitable).

```
r.mapcalc tank.use = “tank.en.route + suitable/100"
```

Adjust the values in this map to represent time in cells per 1000 hours of training.

```
tmp=`r.stats -c tank.use | sed -e s"/ / * /" | bc | \
awk 'BEGIN {printf "0\n"} {print $0 "+"} END {printf "p\n"}' | dc`
r.mapcalc "tank.time = 3600000 * tank.use/$tmp"
```

## Generate Noise Annoyance Map

Presume tank noise carries at a level that annoys the Grouse over 5 cells (150 meters). The following filter operation presumes such a noise annoyance attenuation and sums up the total noise at each cell based on the surrounding cells.

```
cat > /tmp/filter << EOF
TITLE tank noise index
MATRIX 7
0 0 0 1 0 0 0
0 1 2 3 2 1 0
0 2 3 4 3 2 0
1 3 4 5 4 3 1
0 2 3 4 3 2 0
0 1 2 3 2 1 0
0 0 0 1 0 0 0
DIVISOR 69
TYPE P
EOF
r.mfilter tank.time out=tank.noise filter=/tmp/filter
rm /tmp/filter
```

Normalize noise annoyance map to range 0-100

```
r.rescale tank.noise out=tank.noise2 to=0,100
r.mapcalc tank.noise=tank.noise2
```

# Appendix C: STELLA Equations

The following equations, output by the STELLA software, represent the model documented in this paper. These equations were parsed by the translation software to run in conjunction with the immigration algorithm within a parallel processing environment.

```
BROOD_DESIRE = IF (F_SG_EGGS_1+F_SG_EGGS_2 > 0) THEN 0.1
ELSE IF (F_SG_JUVENILES > 0) THEN 0.4
ELSE 1
```

**DOCUMENT:** Units = 0-1 factor representing relative desire to remain in the current location based on maternal instincts.

```
DESIRE_TO_MIGRATE = (1-GENERAL_ATTRACTION) * MAX(0.0,NESTING_DESIRE *
BROOD_DESIRE - NOISE_FACTOR)
GENERAL_ATTRACTION = SNOW_COVER_FACTOR*SAGE_BRUSH_FACTOR*NOISE_FACTOR
```

**DOCUMENT:** Units = 0-1 index measuring the attraction of a cell to the Grouse. 0 = unattractive and 1 = attractive.

```
LEK_ATTRACTION = CV_LEK_DIST*MALES_ON_LEK
```

**DOCUMENT:** This equation presumes that attractiveness increases as the noise from the Lek increases - which itself is simply presumed to be a linear function of the number of males on the lek and the distance to the lek. This ignores any internal driving motivation based on hormonal states within the females.

```
MALES_ON_LEK = GRAPH(CV_WEEK)
(4.00, 0.00), (5.00, 9.50), (6.00, 26.0), (7.00, 54.5), (8.00, 82.5), (9.00, 100), (10.0, 68.0), (11.0, 99.5),
(12.0, 86.5), (13.0, 53.0), (14.0, 24.5), (15.0, 10.0), (16.0, 0.00)
```

**DOCUMENT:** Units = Total number of males expected on the lek.

```
NESTING_DESIRE = GRAPH(F_SG_PREGNANT)
(0.00, 1.00), (1.00, 1.00), (2.00, 0.915), (3.00, 0.705), (4.00, 0.505), (5.00, 0.38), (6.00, 0.29), (7.00, 0.225),
(8.00, 0.205), (9.00, 0.185), (10.0, 0.185)
```

**DOCUMENT:** Units = 0-1 factor representing the desire of the female to remain in her current position based on the point she is in her pregnancy.

NOISE_FACTOR = GRAPH(CVH_NOISE_INDEX)
(0.00, 1.00), (0.1, 0.97), (0.2, 0.895), (0.3, 0.65), (0.4, 0.36), (0.5, 0.185), (0.6, 0.075), (0.7, 0.045), (0.8, 0.025), (0.9, 0.01), (1, 0.00)

**DOCUMENT:** Units = 0-1 factor representing noise as a function of military exercises in the cell.

SAGE_BRUSH_FACTOR = GRAPH(CVP_SAGEBRUSH)
(0.00, 0.1), (3.00, 0.13), (6.00, 0.2), (9.00, 0.465), (12.0, 0.775), (15.0, 0.905), (18.0, 0.96), (21.0, 0.985), (24.0, 0.995), (27.0, 0.995), (30.0, 1.00)

**DOCUMENT:** Units = 0-1 factor representing migration desire as a function of sage brush densities.

SNOW_COVER_FACTOR = GRAPH(CVS_SNOW)
(0.00, 1.00), (1.80, 0.96), (3.60, 0.825), (5.40, 0.465), (7.20, 0.29), (9.00, 0.195), (10.8, 0.135), (12.6, 0.09), (14.4, 0.05), (16.2, 0.025), (18.0, 0.00)

**DOCUMENT:** Units = 0-1 index measuring the desire to migrate as a function of snow depth.

CVH_COMPACTION = SOIL_COMPACTION_INDEX

**DOCUMENT:** Units= percent of land compacted

CVH_NOISE_INDEX = NOISE_INDEX

**DOCUMENT:** Units = ranges from 0 (no noise) to 1 (full noise impact)

CVH_VEG_DAMAGE_INDEX = VEG_DAMAGE

**DOCUMENT:** Units = value that ranges from 0 (no damage) to 1000 (complete damage).

CVP_AGROPYRON = AG_COVER

**DOCUMENT:** Units = percent cover of *Agropyron spicatum*

CVP_FORB = FB_COVER

**DOCUMENT:** Units = percent cover of forbs

CVP_SAGEBRUSH = SB_COVER

**DOCUMENT:** Units = percent cover of sagebrush

CVS_SNOW = SM_SNOW_PAC

**DOCUMENT:** Units = millimeters of water stored in the snow.

CVS_SOIL_MOISTURE = AVAILABLE_SOIL_MOISTURE

**DOCUMENT:** Units = millimeters

CVS_TEMPERATURE = SM_TEMPERATURE

CV_ASPECT = 180

**DOCUMENT:** Units = degrees

CV_ELEVATION = 800

**DOCUMENT:** Units = meters above sea level

CV_ENCAMP_HR = 2000

**DOCUMENT:** Units = personnel encampment hours per 100 hectares per exercise.

CV_LEK_DIST = 100

**DOCUMENT:** Units = meters

CV_SLOPE = 1

**DOCUMENT:** Units = degrees of incline

CV_SOIL_PERM = 0

**DOCUMENT:** Soil permeability in ? units

CV_TRAK_VEH_HR = 10

**DOCUMENT:** Units = moving tracked vehicle hours per 100 hectares per exercise

CV_TROOP_HR = 100

**DOCUMENT:** Units = active troop hours per 100 hectares per exercise

CV_UTRAK_VEH_HR = 10

**DOCUMENT:** Units = moving untracked vehicle hours per 100 hectares per exercise

CV_WATER_HOLDING_CAPACITY = 45.0

**DOCUMENT:** Units = millimeters of water in the first 15 centimeters of soil with 100% saturation.

CV_WEEK = MOD(TIME,52)

**DOCUMENT:** Units = weeks

CV_WETNESS_INDEX = 1.0

**DOCUMENT:** Units = constant

FIRE_DAMAGE = 0

**DOCUMENT:** Units = 1 (fire) or 0 (no fire)

TEST_AGROPYRON = GRAPH(MOD(TIME,52))
(0.00, 30.0), (1.00, 30.0), (2.00, 30.0), (3.00, 30.0), (4.00, 30.0), (5.00, 30.0), (6.00, 30.0), (7.00, 30.0), (8.00, 30.0), (9.00, 30.0), (10.0, 30.0), (11.0, 29.9), (12.0, 29.9), (13.0, 29.9), (14.0, 29.9), (15.0, 29.9), (16.0, 29.9), (17.0, 29.9), (18.0, 29.9), (19.0, 29.9), (20.0, 29.9), (21.0, 29.9), (22.0, 29.9), (23.0, 29.9), (24.0, 29.9), (25.0, 29.9), (26.0, 29.9), (27.0, 29.9), (28.0, 29.9), (29.0, 29.9), (30.0, 29.9), (31.0, 29.9), (32.0, 29.9), (33.0, 29.9), (34.0, 29.9), (35.0, 29.9), (36.0, 29.9), (37.0, 29.9), (38.0, 29.9), (39.0, 29.9), (40.0, 29.9), (41.0, 29.9), (42.0, 29.9), (43.0, 29.9), (44.0, 29.9), (45.0, 29.8), (46.0, 29.8), (47.0, 29.8), (48.0, 29.8), (49.0, 29.8), (50.0, 29.8), (51.0, 29.8)

**DOCUMENT:** Units = Time vs. cover

TEST_COMPACTION = GRAPH(MOD(TIME,52))
(0.00, 0.00), (1.00, 0.00), (2.00, 0.00), (3.00, 0.00), (4.00, 0.00), (5.00, 0.00), (6.00, 0.00), (7.00, 0.00), (8.00, 0.00), (9.00, 0.00), (10.0, 0.00), (11.0, 0.71), (12.0, 0.71), (13.0, 0.7), (14.0, 0.69), (15.0, 0.69), (16.0, 0.68), (17.0, 0.67), (18.0, 0.67), (19.0, 0.66), (20.0, 0.65), (21.0, 0.65), (22.0, 0.64), (23.0, 0.63), (24.0, 0.63), (25.0, 0.62), (26.0, 0.61), (27.0, 0.61), (28.0, 0.6), (29.0, 0.6), (30.0, 0.59), (31.0, 0.58), (32.0, 0.58), (33.0, 0.57), (34.0, 0.57), (35.0, 0.56), (36.0, 0.56), (37.0, 0.55), (38.0, 0.54), (39.0, 0.54), (40.0, 0.53), (41.0, 0.53), (42.0, 0.52), (43.0, 0.52), (44.0, 0.51), (45.0, 0.51), (46.0, 0.5), (47.0, 0.5), (48.0, 0.49), (49.0, 0.49), (50.0, 0.48), (51.0, 0.48)

**DOCUMENT:** Units = time vs. compaction

TEST_FORB = GRAPH(MOD(TIME,52))
(0.00, 6.00), (1.00, 5.99), (2.00, 5.98), (3.00, 5.97), (4.00, 5.95), (5.00, 5.94), (6.00, 5.93), (7.00, 5.92), (8.00, 5.91), (9.00, 5.90), (10.0, 5.89), (11.0, 5.87), (12.0, 5.86), (13.0, 5.85), (14.0, 5.84), (15.0, 5.82), (16.0, 5.81), (17.0, 5.80), (18.0, 5.79), (19.0, 5.78), (20.0, 5.77), (21.0, 5.76), (22.0, 5.75), (23.0, 5.74), (24.0, 5.72), (25.0, 5.71), (26.0, 5.70), (27.0, 5.69), (28.0, 5.68), (29.0, 5.67), (30.0, 5.66), (31.0, 5.65), (32.0, 5.64), (33.0, 5.63), (34.0, 5.61), (35.0, 5.60), (36.0, 5.59), (37.0, 5.58), (38.0, 5.57), (39.0, 5.56), (40.0, 5.55), (41.0, 5.54), (42.0, 5.53), (43.0, 5.52), (44.0, 5.51), (45.0, 5.50), (46.0, 5.49), (47.0, 5.48), (48.0, 5.47), (49.0, 5.46), (50.0, 5.44), (51.0, 5.43)

**DOCUMENT:** Units = time vs. cover

TEST_NOISE = GRAPH(MOD(TIME,52))
(0.00, 0.00), (1.00, 0.00), (2.00, 0.00), (3.00, 0.00), (4.00, 0.00), (5.00, 0.00), (6.00, 0.00), (7.00, 0.00), (8.00, 0.00), (9.00, 0.00), (10.0, 1.00), (11.0, 0.00), (12.0, 0.00), (13.0, 0.00), (14.0, 0.00), (15.0, 0.00), (16.0, 0.00), (17.0, 0.00), (18.0, 0.00), (19.0, 0.00), (20.0, 0.00), (21.0, 0.00), (22.0, 0.00), (23.0, 0.00), (24.0, 0.00), (25.0, 0.00), (26.0, 0.00), (27.0, 0.00), (28.0, 0.00), (29.0, 0.00), (30.0, 0.00), (31.0, 0.00), (32.0, 0.00), (33.0, 0.00), (34.0, 0.00), (35.0, 0.00), (36.0, 0.00), (37.0, 0.00), (38.0, 0.00), (39.0, 0.00), (40.0, 0.00), (41.0, 0.00), (42.0, 0.00), (43.0, 0.00), (44.0, 0.00), (45.0, 0.00), (46.0, 0.00), (47.0, 0.00), (48.0, 0.00), (49.0, 0.00), (50.0, 0.00), (51.0, 0.00)

**DOCUMENT:** Units = time vs. noise

TEST_SAGEBRUSH = GRAPH(MOD(TIME,52))
(0.00, 18.0), (1.00, 18.0), (2.00, 18.0), (3.00, 18.0), (4.00, 18.0), (5.00, 18.0), (6.00, 18.0), (7.00, 17.9),
(8.00, 17.9), (9.00, 17.9), (10.0, 17.9), (11.0, 17.9), (12.0, 17.9), (13.0, 17.9), (14.0, 17.9), (15.0, 17.9),
(16.0, 17.9), (17.0, 17.9), (18.0, 17.9), (19.0, 17.9), (20.0, 17.9), (21.0, 17.8), (22.0, 17.8), (23.0, 17.8),
(24.0, 17.8), (25.0, 17.8), (26.0, 17.8), (27.0, 17.8), (28.0, 17.8), (29.0, 17.8), (30.0, 17.8), (31.0, 17.8),
(32.0, 17.8), (33.0, 17.8), (34.0, 17.8), (35.0, 17.7), (36.0, 17.7), (37.0, 17.7), (38.0, 17.7), (39.0, 17.7),
(40.0, 17.7), (41.0, 17.7), (42.0, 17.7), (43.0, 17.7), (44.0, 17.7), (45.0, 17.7), (46.0, 17.7), (47.0, 17.7),
(48.0, 17.7), (49.0, 17.6), (50.0, 17.6), (51.0, 17.6)

**DOCUMENT:** Units = time vs. cover

TEST_SNOW = GRAPH(MOD(TIME,52))
(0.00, 30.0), (1.00, 29.4), (2.00, 28.4), (3.00, 26.7), (4.00, 24.8), (5.00, 22.2), (6.00, 19.8), (7.00, 17.4),
(8.00, 13.9), (9.00, 10.7), (10.0, 6.65), (11.0, 2.08), (12.0, 0.33), (13.0, 0.03), (14.0, 0.00), (15.0, 0.00),
(16.0, 0.00), (17.0, 0.00), (18.0, 0.00), (19.0, 0.00), (20.0, 0.00), (21.0, 0.00), (22.0, 0.00), (23.0, 0.00),
(24.0, 0.00), (25.0, 0.00), (26.0, 0.00), (27.0, 0.00), (28.0, 0.00), (29.0, 0.00), (30.0, 0.00), (31.0, 0.00),
(32.0, 0.00), (33.0, 0.00), (34.0, 0.00), (35.0, 0.00), (36.0, 0.00), (37.0, 0.00), (38.0, 0.00), (39.0, 0.00),
(40.0, 0.00), (41.0, 0.00), (42.0, 0.00), (43.0, 0.00), (44.0, 0.00), (45.0, 0.00), (46.0, 0.00), (47.0, 0.00),
(48.0, 6.29), (49.0, 12.6), (50.0, 19.0), (51.0, 25.5)

**DOCUMENT:** Units = time vs snow

TEST_SOIL_MOISTURE = GRAPH(MOD(TIME,52))
(0.00, 10.0), (1.00, 7.18), (2.00, 5.64), (3.00, 4.55), (4.00, 3.75), (5.00, 3.31), (6.00, 3.37), (7.00, 4.05),
(8.00, 5.55), (9.00, 7.71), (10.0, 9.99), (11.0, 11.8), (12.0, 9.46), (13.0, 7.26), (14.0, 5.72), (15.0, 4.61),
(16.0, 3.78), (17.0, 3.12), (18.0, 2.59), (19.0, 2.16), (20.0, 1.80), (21.0, 1.51), (22.0, 1.27), (23.0, 1.06),
(24.0, 0.89), (25.0, 0.75), (26.0, 0.63), (27.0, 0.53), (28.0, 0.45), (29.0, 0.38), (30.0, 0.32), (31.0, 0.27),
(32.0, 0.23), (33.0, 0.19), (34.0, 0.16), (35.0, 0.14), (36.0, 0.12), (37.0, 0.1), (38.0, 0.08), (39.0, 0.07), (40.0,
0.06), (41.0, 0.05), (42.0, 0.04), (43.0, 0.04), (44.0, 0.03), (45.0, 0.03), (46.0, 0.02), (47.0, 0.02), (48.0,
0.02), (49.0, 0.01), (50.0, 0.01), (51.0, 0.01)

TEST_TEMPERATURE = GRAPH(MOD(TIME,52))
(0.00, -2.50), (1.00, -1.65), (2.00, -0.8), (3.00, 0.05), (4.00, 0.9), (5.00, 1.90), (6.00, 2.94), (7.00, 3.99),
(8.00, 5.03), (9.00, 6.11), (10.0, 7.22), (11.0, 8.33), (12.0, 9.44), (13.0, 10.5), (14.0, 11.6), (15.0, 12.6),
(16.0, 13.7), (17.0, 14.7), (18.0, 15.5), (19.0, 16.3), (20.0, 17.0), (21.0, 17.8), (22.0, 18.6), (23.0, 19.5),
(24.0, 20.4), (25.0, 21.2), (26.0, 21.5), (27.0, 21.2), (28.0, 20.9), (29.0, 20.6), (30.0, 20.1), (31.0, 19.1),
(32.0, 18.2), (33.0, 17.2), (34.0, 16.3), (35.0, 14.9), (36.0, 13.4), (37.0, 12.0), (38.0, 10.6), (39.0, 8.99),
(40.0, 7.28), (41.0, 5.57), (42.0, 3.86), (43.0, 2.61), (44.0, 1.84), (45.0, 1.07), (46.0, 0.3), (47.0, -0.41),
(48.0, -0.93), (49.0, -1.45), (50.0, -1.98), (51.0, -2.50)

TEST_VEG_DAMAGE = GRAPH(MOD(TIME,52))
(0.00, 0.00), (1.00, 0.00), (2.00, 0.00), (3.00, 0.00), (4.00, 0.00), (5.00, 0.84), (6.00, 0.00), (7.00, 0.00),
(8.00, 0.00), (9.00, 0.00), (10.0, 1.00), (11.0, 0.00), (12.0, 0.00), (13.0, 0.00), (14.0, 0.00), (15.0, 0.00),
(16.0, 0.00), (17.0, 0.00), (18.0, 0.00), (19.0, 0.00), (20.0, 0.00), (21.0, 0.00), (22.0, 0.00), (23.0, 0.00),
(24.0, 0.00), (25.0, 0.00), (26.0, 0.00), (27.0, 0.00), (28.0, 0.00), (29.0, 0.00), (30.0, 0.00), (31.0, 0.00),
(32.0, 0.00), (33.0, 0.00), (34.0, 0.00), (35.0, 0.00), (36.0, 0.00), (37.0, 0.00), (38.0, 0.00), (39.0, 0.00),
(40.0, 0.00), (41.0, 0.00), (42.0, 0.00), (43.0, 0.00), (44.0, 0.00), (45.0, 0.00), (46.0, 0.00), (47.0, 0.00),
(48.0, 0.00), (49.0, 0.00), (50.0, 0.00), (51.0, 0.00)

Training_Schedule = GRAPH(MOD(TIME,52))
(0.00, 0.00), (1.00, 0.00), (2.00, 1.00), (3.00, 0.00), (4.00, 0.00), (5.00, 0.00), (6.00, 1.00), (7.00, 0.00), (8.00, 0.00), (9.00, 0.00), (10.0, 0.00), (11.0, 0.00), (12.0, 0.00), (13.0, 0.00), (14.0, 0.00), (15.0, 0.00), (16.0, 0.00), (17.0, 0.00), (18.0, 0.00), (19.0, 0.00), (20.0, 0.00), (21.0, 0.00), (22.0, 0.00), (23.0, 0.00), (24.0, 0.00), (25.0, 0.00), (26.0, 0.00), (27.0, 0.00), (28.0, 0.00), (29.0, 0.00), (30.0, 0.00), (31.0, 0.00), (32.0, 0.00), (33.0, 0.00), (34.0, 0.00), (35.0, 0.00), (36.0, 0.00), (37.0, 0.00), (38.0, 0.00), (39.0, 0.00), (40.0, 0.00), (41.0, 0.00), (42.0, 0.00), (43.0, 0.00), (44.0, 0.00), (45.0, 0.00), (46.0, 0.00), (47.0, 0.00), (48.0, 0.00), (49.0, 0.00), (50.0, 0.00), (51.0, 0.00)

**DOCUMENT:** Units = 1(training regime) or 0 (no training)

SOIL_COMPACTION_INDEX(t) = SOIL_COMPACTION_INDEX(t - dt) + (NEW_COMPACTION - UNCOMPACTION) * dt
INIT SOIL_COMPACTION_INDEX = 0

**DOCUMENT:** Units = Percent land compacted.

NEW_COMPACTION = COMPACTION * (1 - SOIL_COMPACTION_INDEX)

**DOCUMENT:** Units = percent of land to be compacted.

UNCOMPACTION = SOIL_COMPACTION_INDEX * 0.01

**DOCUMENT:** Units = percent land uncompacted every dt

CV_TRAK_VEH_NOISE = 10

**DOCUMENT:** Units = Tracked vehicle hours in use per 100 hectares per exercise

NOISE_COEF_C = 0.5

**DOCUMENT:** Units = troop-hr noise units

NOISE_COEF_T = 1

**DOCUMENT:** units = troop-hr noise units

NOISE_COEF_TV = 100

**DOCUMENT:** units = troop-hr noise units

NOISE_COEF_UTV = 50

**DOCUMENT:** Units = troop-hr noise units

SOIL_COMPACT_COEF_C = 0.001

**DOCUMENT:** Units = hectares compacted per troop-hour of bivouacing.

SOIL_COMPACT_COEF_T = 0.0001

**DOCUMENT:** Units = hectares compacted per troop-hour of training

SOIL_COMPACT_COEF_TV = 0.15

**DOCUMENT:** Units = hectares compacted per tracked-vehicle-hour of training

SOIL_COMPACT_COEF_UTV = 0.10

**DOCUMENT:** Units = hectares compacted per untracked-vehicle-hour of training

SOIL_COM_C_MULT = CV_ENCAMP_HR * SOIL_COMPACT_COEF_C / 100

**DOCUMENT:** Units = area damaged per hectare

SOIL_COM_TV_MULT = SOIL_COMPACT_COEF_TV * CV_TRAK_VEH_HR / 100

**DOCUMENT:** Units = area damaged per hectare

SOIL_COM_T_MULT = SOIL_COMPACT_COEF_T * CV_TROOP_HR / 100

**DOCUMENT:** Units = area damaged per hectare

SOIL_COM_UTV_MULT = SOIL_COMPACT_COEF_UTV * CV_UTRAK_VEH_HR / 100

**DOCUMENT:** Units = area damaged per hectare

VEG_DAMAGE_COEF_C = 1

**DOCUMENT:** Units = vegetation impact units per troop-hour of training

VEG_DAMAGE_COEF_T = 1

**DOCUMENT:** Units = vegetation impact units per troop-hour of training

VEG_DAMAGE_COEF_TV = 1000

**DOCUMENT:** Units = vegetation impact units per troop-hour of training

VEG_DAMAGE_COEF_UTV = 500

**DOCUMENT:** Units = vegetation impact units per troop-hour of training

VEG_DAMAGE_C_MULT = CV_ENCAMP_HR * VEG_DAMAGE_COEF_C / 100

**DOCUMENT:** Units = vegetation impact units per hectare

VEG_DAMAGE_TV_MULT = CV_TRAK_VEH_HR * VEG_DAMAGE_COEF_TV / 100

**DOCUMENT:** Units = vegetation impact units per hectare

VEG_DAMAGE_T_MULT = VEG_DAMAGE_COEF_T * CV_TROOP_HR / 100

**DOCUMENT:** Units = vegetation impact units per hectare

VEG_DAMAGE_UTV_MULT = VEG_DAMAGE_COEF_UTV * CV_UTRAK_VEH_HR / 100

**DOCUMENT:** Units = vegetation impact units per hectare

COMPACTION = GRAPH(Training_Schedule * (SOIL_COM_TV_MULT + SOIL_COM_UTV_MULT + SOIL_COM_C_MULT + SOIL_COM_T_MULT) / 10000)
(0.00, 0.00), (1.00, 0.36), (2.00, 0.55), (3.00, 0.7), (4.00, 0.8), (5.00, 0.88), (6.00, 0.93), (7.00, 0.97), (8.00, 0.99), (9.00, 0.995), (10.0, 1.00)

**DOCUMENT:** Units = 0 -1 index measuring total compaction of the system per exercise.

NOISE_INDEX = GRAPH(Training_Schedule*CV_TRAK_VEH_NOISE)
(0.00, 0.00), (10.0, 0.235), (20.0, 0.44), (30.0, 0.59), (40.0, 0.695), (50.0, 0.78), (60.0, 0.85), (70.0, 0.92), (80.0, 0.97), (90.0, 0.995), (100, 1.00)

**DOCUMENT:** Units = 0-100 percent index per exercise

VEG_DAMAGE = GRAPH(Training_Schedule * (VEG_DAMAGE_UTV_MULT + VEG_DAMAGE_TV_MULT + VEG_DAMAGE_C_MULT + VEG_DAMAGE_T_MULT))
(0.00, 0.00), (100, 0.365), (200, 0.555), (300, 0.705), (400, 0.795), (500, 0.865), (600, 0.92), (700, 0.96), (800, 0.98), (900, 0.995), (1000, 1.00)

**DOCUMENT:**

1. Compute HA-damaged/HA/Exercise by summing the following:
   Troop-hours/HA/Exercise * HA-damaged/Troop-hour
   TV-hours/HA/Exercise * HA-damaged/TV-hour
   UTV-hours/HA/Exercise * HA-damaged/UTV-hour
   Bivouac-hours/HA/Exercise * HA-damaged/Bivouac-hour
2. Compute HA-damaged/HA by multiplying:
   HA-damaged/HA/Exercise * Exercises
3. Compute noise index by sending HA-damaged/HA through graph

AG_COVER(t) = AG_COVER(t - dt) + (NEW_AG_COVER - AG_FIRE - AG_NATURAL_DEATH - AG_HUMAN_IMPACT - AG_CONSUMPTION) * dt
INIT AG_COVER = 30

**DOCUMENT:** Units = percent cover

NEW_AG_COVER = MIN(NEW_AG,AG_COVER-59)

**DOCUMENT:** Units = percent cover

AG_FIRE = IF FIRE_DAMAGE=1 THEN AG_COVER ELSE 0

**DOCUMENT:** Units = percent agropyron cover

AG_NATURAL_DEATH = AG_COVER*.005/52

**DOCUMENT:** Units = percent cover dying

AG_HUMAN_IMPACT = IF( CVH_VEG_DAMAGE_INDEX > 0) THEN AG_COVER-(CVH_VEG_DAMAGE_INDEX*AG_COVER) ELSE 0

**DOCUMENT:** Units = percent cover left after a human impact

AG_CONSUMPTION = AG_COVER*0

**DOCUMENT:** Units = percent cover lost due to consumption

FB_COVER(t) = FB_COVER(t - dt) + (NEW_FB_COVER - FB_FIRE - FB_NAT_DEATH - FB_HUMAN_IMPACT - FB_CONSUMPTION) * dt
INIT FB_COVER = 6

**DOCUMENT:** Units = percent cover forbs

NEW_FB_COVER = MIN(NEW_FB,FB_COVER-59)

**DOCUMENT:** Units = percent cover

FB_FIRE = IF FIRE_DAMAGE=1 THEN FB_COVER ELSE 0

**DOCUMENT:** Units = percent cover consumed per fire

FB_NAT_DEATH = FB_COVER*.1/52

**DOCUMENT:** Units = percent cover

FB_HUMAN_IMPACT = IF (CVH_VEG_DAMAGE_INDEX>0) THEN FB_COVER-(CVH_VEG_DAMAGE_INDEX*FB_COVER) ELSE 0

**DOCUMENT:** Units = 0 to 1 index representing percent cover.

FB_CONSUMPTION = FB_COVER*0

**DOCUMENT:** Units = percent cover lost due to consumption

SB_COVER(t) = SB_COVER(t - dt) + (NEW_SB_COVER - SB_FIRE - SB_NAT_DEATH - SB_HUMAN_IMPACT - SB_CONSUMPTION) * dt
INIT SB_COVER = 18

**DOCUMENT:** Units = percent cover

NEW_SB_COVER = MIN(NEW_SB,SB_COVER-25)

**DOCUMENT:** Units = percent cover

SB_FIRE = IF FIRE_DAMAGE=1 THEN SB_COVER ELSE 0

**DOCUMENT:** Units = percent cover burned

SB_NAT_DEATH = SB_COVER*.02/52

**DOCUMENT:** Units = percent cover

SB_HUMAN_IMPACT = IF (CVH_VEG_DAMAGE_INDEX>0) THEN SB_COVER-(CVH_VEG_DAMAGE_INDEX*SB_COVER) ELSE 0

**DOCUMENT:** Units = percent cover

SB_CONSUMPTION = SB_COVER*0

**DOCUMENT:** Units = percent cover

AG_POST_FIRE_COVER = DELAY(AG_REGROWTH,78)
**DOCUMENT:** Units = percent cover

AG_REGROWTH = if FIRE_DAMAGE=1 THEN 0.05*TOTAL_COVER ELSE 0
**DOCUMENT:** Units = percent cover

FB_POST_FIRE_COVER = DELAY(FB_REGROWTH,78)
**DOCUMENT:** Units = percent cover

FB_REGROWTH = if FIRE_DAMAGE=1 THEN 0.05*TOTAL_COVER ELSE 0
**DOCUMENT:** Units = percent cover

NEW_AG = IF (CVS_TEMPERATURE_12.0) AND
(CVS_TEMPERATURE_22.0) AND
(CVS_SOIL_MOISTURE_8.0) AND
(CVS_SOIL_MOISTURE_10.00) AND
(CV_WEEK_15) AND
(CV_WEEK_25) THEN
AG_COVER *(AG_ASPECT_MOD* 0.02) +AG_POST_FIRE_COVER
ELSE 0
**DOCUMENT:** Units = percent cover added per dt

NEW_FB = IF (CVS_TEMPERATURE_12.0) AND
(CVS_TEMPERATURE_22.0) AND
(CVS_SOIL_MOISTURE_8.0) AND
(CVS_SOIL_MOISTURE_10.00) AND
(CV_WEEK_15) AND
(CV_WEEK_25) THEN
FB_COVER *(FB_ASPECT_MOD* 0.02) +FB_POST_FIRE_COVER
ELSE 0
**DOCUMENT:** Units = percent cover added per dt

NEW_SB = IF (CVS_TEMPERATURE_12.0) AND
(CVS_TEMPERATURE_22.0) AND
(CVS_SOIL_MOISTURE_8.0) AND
(CVS_SOIL_MOISTURE_10.00) AND
(CV_WEEK_15) AND
(CV_WEEK_25) THEN
SB_COVER *(SB_ASPECT_MOD* 0.02) +SB_POST_FIRE_COVER
ELSE 0
**DOCUMENT:** Units = percent cover added per dt

SB_POST_FIRE_COVER = DELAY(SB_REGROWTH,78)
**DOCUMENT:** Units = percent cover

```
SB_REGROWTH = if FIRE_DAMAGE=1 THEN 0.05*TOTAL_COVER ELSE 0
```

**DOCUMENT:** Units = percent cover

```
TOTAL_COVER = MIN(100,AG_COVER+FB_COVER+SB_COVER)
```

**DOCUMENT:** Units = percent cover of all vegetation

```
AG_ASPECT_MOD = GRAPH(CV_ASPECT)
(0.00, 0.6), (36.4, 0.605), (72.8, 0.61), (109, 0.635), (146, 0.71), (182, 0.95), (218, 1.00), (255, 1.00), (291,
0.87), (328, 0.725), (364, 0.6)
```

**DOCUMENT:** Units = degrees

```
FB_ASPECT_MOD = GRAPH(CV_ASPECT)
(0.00, 0.6), (36.4, 0.605), (72.8, 0.61), (109, 0.635), (146, 0.71), (182, 0.95), (218, 1.00), (255, 1.00), (291,
0.87), (328, 0.725), (364, 0.6)
```

**DOCUMENT:** Units = degrees

```
SB_ASPECT_MOD = GRAPH(CV_ASPECT)
(0.00, 0.6), (36.4, 0.605), (72.8, 0.61), (109, 0.635), (146, 0.71), (182, 0.95), (218, 1.00), (255, 1.00), (291,
0.87), (328, 0.725), (364, 0.6)
```

**DOCUMENT:** Units = degrees

```
F_SG_ADULTS(t) = F_SG_ADULTS(t - dt) + (SURVIVAL_2 - F_ADULT_DEATHS) * dt
INIT F_SG_ADULTS = 1{female Grouse}
```

**DOCUMENT:** Units = female adults per hectare (?)

```
SURVIVAL_2 = IF (CV_WEEK = 0) THEN F_SG_JUVENILES/DT ELSE 0
```

**DOCUMENT:** Units = surviving female juveniles

```
F_ADULT_DEATHS = IF ( Random (0.0,1.0) < (F_ADULT_DEATH_1 - INT(F_ADULT_DEATH_1)))
THEN (1/DT) * (INT (F_ADULT_DEATH_1) +1)
ELSE (1/DT) * ( INT (F_ADULT_DEATH_1))
```

**DOCUMENT:** Units = number of female adult deaths

```
F_SG_EGGS_1(t) = F_SG_EGGS_1(t - dt) + (EGG_LAYING - EGG_SURV_1 - EGG_DEATH_1 -
ABANDON_DEATH_1) * dt
INIT F_SG_EGGS_1 = 0
```

**DOCUMENT:** Units = number of eggs

```
EGG_LAYING = IF (Random(0.0,1.0) < (EGG_LAY - INT(EGG_LAY)))
THEN (1/DT) * (INT(EGG_LAY) + 1)
ELSE (1/DT) * (INT(EGG_LAY))
```

**DOCUMENT:** Units = number of eggs hatched per pair of Grouse

```
EGG_SURV_1 = IF (Random(0.0,1.0) < (DT * F_SG_EGGS_1 - INT(DT * F_SG_EGGS_1)))
THEN (1/(DT * DT)) * (INT(DT * F_SG_EGGS_1) + 1)
ELSE (1/(DT * DT)) * (INT(DT * F_SG_EGGS_1) )
```

**DOCUMENT:** Units = number of eggs

```
EGG_DEATH_1 = IF (DT * Random(0.0,1.0) < (EGG_DEATH_A - INT(EGG_DEATH_A)))
THEN (1/DT) * (INT(EGG_DEATH_A) + 1)
ELSE (1/DT) * (INT(EGG_DEATH_A))
```

**DOCUMENT:** Units = number of eggs

```
ABANDON_DEATH_1 = IF (F_SG_ADULTS = 0) THEN 1/DT * F_SG_EGGS_1 ELSE 0
```

**DOCUMENT:** Units = number of eggs

```
F_SG_EGGS_2(t) = F_SG_EGGS_2(t - dt) + (EGG_SURV_1 - EGG_SURV_2 - EGG_DEATH_2 -
ABANDON_DEATH_2) * dt
INIT F_SG_EGGS_2 = 0
```

**DOCUMENT:** Units = number of eggs

```
EGG_SURV_1 = IF (Random(0.0,1.0) < (DT * F_SG_EGGS_1 - INT(DT * F_SG_EGGS_1)))
THEN (1/(DT * DT)) * (INT(DT * F_SG_EGGS_1) + 1)
ELSE (1/(DT * DT)) * (INT(DT * F_SG_EGGS_1) )
```

**DOCUMENT:** Units = number of eggs

```
EGG_SURV_2 = IF (Random(0.0,1.0) < (DT * F_SG_EGGS_2 - INT(DT * F_SG_EGGS_2)))
THEN (1/(DT * DT)) * (INT(DT * F_SG_EGGS_2) + 1)
ELSE (1/(DT * DT)) * (INT(DT * F_SG_EGGS_2) )
```

**DOCUMENT:** Units = number of eggs

```
EGG_DEATH_2 = IF (DT * Random(0.0,1.0) < (EGG_DEATH_B - INT(EGG_DEATH_B)))
THEN (1/DT) * (INT(EGG_DEATH_B) + 1)
ELSE (1/DT) * (INT(EGG_DEATH_B))
ABANDON_DEATH_2 = IF (F_SG_ADULTS = 0) THEN 1/DT * F_SG_EGGS_2 ELSE 0
F_SG_JUVENILES(t) = F_SG_JUVENILES(t - dt) + (EGG_SURV_2 - F_JUVENILE_DEATH -
SURVIVAL_2) * dt
INIT F_SG_JUVENILES = 0
EGG_SURV_2 = IF (Random(0.0,1.0) < (DT * F_SG_EGGS_2 - INT(DT * F_SG_EGGS_2)))
THEN (1/(DT * DT)) * (INT(DT * F_SG_EGGS_2) + 1)
ELSE (1/(DT * DT)) * (INT(DT * F_SG_EGGS_2) )
```

**DOCUMENT:** Units = number of eggs

```
F_JUVENILE_DEATH = IF (DT * Random(0.0,1.0) < (F_JUVENILE_DEATH_1 -
INT(F_JUVENILE_DEATH_1)))
THEN (1/DT) * (INT(F_JUVENILE_DEATH_1) + 1)
ELSE (1/DT) * (INT(F_JUVENILE_DEATH_1))
```

**DOCUMENT:** Units = number of female juvenile deaths

SURVIVAL_2 = IF (CV_WEEK = 0) THEN F_SG_JUVENILES/DT ELSE 0

**DOCUMENT:** Units = surviving female juveniles

F_SG_PREGNANT(t) = F_SG_PREGNANT(t - dt) + (FERTILIZE - GESTATE) * dt
INIT F_SG_PREGNANT = 0

**DOCUMENT:** Units = number of pregnant female Sage Grouse

FERTILIZE = IF (CV_LEK_DIST = 0 & F_SG_PREGNANT = 0 & F_SG_JUVENILES = 0) THEN
F_SG_ADULTS*MALES_ON_LEK* 0.1
ELSE 0

**DOCUMENT:** Units = number of female Sage Grouse fertilized per dt

GESTATE = DT

**DOCUMENT:** Units = number of female Sage Grouse gestating per dt (?)

ADULT_SURV_FRAC = 0.30

**DOCUMENT:** Units = constant that gives adults surviving.

ADULT_WEEKS = 156

**DOCUMENT:** Units = weeks

EGGS_PER_FEMALE = 6

**DOCUMENT:** Units = eggs

EGG_DEATH_A = IF (EGG_TOTAL = 0) THEN 0
ELSE (F_SG_EGGS_1/EGG_TOTAL) * SG_EGG_DEATH

**DOCUMENT:** Units = number of eggs

EGG_DEATH_B = SG_EGG_DEATH-EGG_DEATH_A

**DOCUMENT:** Units = number of eggs

EGG_LAY = IF (FERTILIZE _ 0) THEN EGGS_PER_FEMALE ELSE 0

**DOCUMENT:** Units = number of eggs

EGG_SURV_FRACTION = 0.38

**DOCUMENT:** Units = fraction of eggs surviving

EGG_TOTAL = F_SG_EGGS_1+F_SG_EGGS_2

**DOCUMENT:** Units = number of eggs

EGG_WEEKS = 2

**DOCUMENT:** Units = weeks

F_ADULT_DEATH_1 = (1-MOD_ADULT_SUR_FRAC)*(F_SG_ADULTS/DT)

**DOCUMENT:** Units = adult deaths

F_JUVENILE_DEATH_1 = DT * (1-MOD_JUV_SUR_FRAC) * F_SG_JUVENILES

**DOCUMENT:** Units = juvenile deaths

F_SG_POPULATION = F_SG_ADULTS+F_SG_JUVENILES

**DOCUMENT:** Units = number of Sage Grouse

JUV_WEEKS = 50

**DOCUMENT:** Units = weeks

MOD_ADULT_SUR_FRAC = EXP(LOGN(ADULT_SURV_FRAC)*(DT/ADULT_WEEKS))

**DOCUMENT:** Units = fraction of adults surviving

MOD_EGG_SUR_FRAC = 1/DT * ( 1 - EXP(LOGN(EGG_SURV_FRACTION)*(DT/EGG_WEEKS)))

**DOCUMENT:** Units = fraction of eggs surviving

MOD_JUV_SUR_FRAC = (EXP(LOGN(JUV_SURV_FRACT)*(DT/JUV_WEEKS)))

**DOCUMENT:** Units = fraction of juveniles surviving

SG_EGG_DEATH = DT * (1-MOD_EGG_SUR_FRAC) * EGG_TOTAL

**DOCUMENT:** Units = number of eggs dying

JUV_SURV_FRACT = GRAPH(CVP_SAGEBRUSH)
(0.00, 0.16), (2.50, 0.16), (5.00, 0.18), (7.50, 0.22), (10.0, 0.27), (12.5, 0.355), (15.0, 0.47), (17.5, 0.6), (20.0, 0.7), (22.5, 0.775), (25.0, 0.8)

**DOCUMENT:** Units = fraction of juveniles surviving

CELL_SIZE = 0.09

**DOCUMENT:** Units = hectares

Hab = on_map

**DOCUMENT:** Units = constant

on_map = 1

**DOCUMENT:** Units = ?

AVAILABLE_SOIL_MOISTURE(t) = AVAILABLE_SOIL_MOISTURE(t - dt) +
(SM_ACTUAL_INFILTRATION - SM_DECREASE) * dt
INIT AVAILABLE_SOIL_MOISTURE = 10

**DOCUMENT:** Units = millimeters of water (stored in the top 15 centimeters)

```
SM_ACTUAL_INFILTRATION = SM_POTENTIAL_INFILTRATION*SM_INFILTRATION_RATE
SM_DECREASE = IF (SM_SWITCH = 1) THEN {IN MILLIMETERS LOST FROM STORAGE}
        SM_ADJUSTED_AE
ELSE
        SM_PE_CONVERSION
```

**DOCUMENT:** This is really the rate of evapotranspiration from the soil in the specific cell being modeled.

```
((-400 + 30*SM_Temp - 0.4*SM_Temp*SM_Temp)*SM_Radiation_Index*SM_Cover)*10
IF SM_Temp > 26.5 THEN (-415.85+(32.42*SM_Temp)-(0.43*SM_Temp^2)) ELSE
SM_Cover*SM_Radiation_Index
IF SOIL_MOISTURE_STORAGE < SM_MAX AND SOIL_MOISTURE_STORAGE > SM_MIN THEN
SM_PE_CONVERSION ELSE IF SOIL_MOISTURE_STORAGE < SM_MIN THEN SM_ADJUSTED_AE
ELSE 0
SM_ADJUSTED_AE
SM_SNOW_PAC(t) = SM_SNOW_PAC(t - dt) + (SM_SNOW - SM_MELT) * dt
INIT SM_SNOW_PAC = 0
```

**DOCUMENT:** Units = millimeters of water in the snow

```
SM_SNOW = IF SM_TEMPERATURE <= 0 THEN SM_RAINFALL_CONVERSION
ELSE 0.00
```

**DOCUMENT:** Units = millimeters of water in the snow

```
SM_MELT = IF SM_TEMPERATURE >= 0
THEN SM_MELT_VS_TEMP * SM_SNOW_PAC
ELSE 0.00
```

**DOCUMENT:** Units = millimeters of water melting from the snow pack

```
SM_A = ((6.75/10.0^7.0) * SM_HEAT^3.00) - ((7.71/10.0^5.0) * SM_HEAT^2.00) + ((1.79/10.0^2.0) *
SM_HEAT) + 0.49
```

**DOCUMENT:**

(0.0638*SM_HEAT^3) - (0.2724*SM_HEAT^2.0) +
(0.0320*SM_HEAT) + 0.49
{ THE FULL EQUATION IS AS FOLLOWS: ((6.75/10)^7.0 *
SM_HEAT^3.00) -((7.71/10)^5.0 * SM_HEAT^2.00) + ((1.79/10)^2.0 *
SM_HEAT) + 0.49}

```
SM_ACTUAL_EVPT = AVAILABLE_SOIL_MOISTURE*SM_PERCENT_LOSS_FROM_STORAGE
```

**DOCUMENT:** Units = millimeters of water

```
SM_ADJUSTED_AE = SM_ACTUAL_EVPT*SM_COVER_INDEX
SM_ASPECT_RECLASS = CV_ASPECT
```

**DOCUMENT:** Units = [constant = 1]

```
SM_AVAILABLE_SOIL_WATER =
(AVAILABLE_SOIL_MOISTURE/CV_WATER_HOLDING_CAPACITY)*100
```

**DOCUMENT:** Units = percent of the potential saturation

SM_BASE_ELEVATION = 800

**DOCUMENT:** Units = meters

SM_COVER_INDEX = 0.55*((LEAF_AREA_INDEX)^0.5)

SM_CRITICAL_PERCENT = 0.70

**DOCUMENT:** Units = percent of maximum holding capacity

SM_HEAT = (12/12) * ((118.44 / 5.0) ^1.514)

**DOCUMENT:** Units = degrees Celsius

SM_MIN = SM_CRITICAL_PERCENT*CV_WATER_HOLDING_CAPACITY

**DOCUMENT:** ****SEE SM_CRITICAL_PERCENT****

SM_NEW_WATER = SM_RAINFALL_CONVERSION-SM_SNOW+SM_MELT

SM_PE_CONVERSION = (SM_POTENTIAL_EVPT/4.3)

{THIS SHOULD BE ZERO WHEN SOIL MOISTURE STORAGE IS LESS THAN THE MINIMUM SPECIFIED IN "SM_MIN" OR WHEN THE SM_STORAGE IS GREATER THAN THAT SPECIFIED IN "SM_MAX"}

SM_POTENTIAL_EVPT = IF SM_TEMPERATURE > 0.00 THEN
16.0 * ((10.0 * (SM_TEMPERATURE / SM_HEAT )) ^ SM_A)
ELSE 0

SM_POTENTIAL_INFILTRATION = SM_COVER_INDEX * CV_WETNESS_INDEX * SM_NEW_WATER * SM_SOIL_TRANSMISSIVITY_INDEX

**DOCUMENT:** Units = millimeters of water

SM_RAINFALL_CONVERSION = SM_RAINFALL_MM/4.3

**DOCUMENT:** Units = millimeters of water per week

SM_SLOPE_RECLASS = CV_SLOPE

**DOCUMENT:** Units = constant [1]

SM_SOIL_TRANSMISSIVITY_INDEX = (1-CVH_COMPACTION)*CV_SOIL_PERM

**DOCUMENT:** THIS WILL BE FOUND BY COMPUTING [ln (Te/Ti)] WHERE Te = AVERAGE TRANSMISSIVITY AND Ti = TRANSMISSIVITY OF THE SPECIFIC CELL. THE INDEX FROM THE HUMAN DISTURBANCE GROUP WILL BE USED TO MODIFY Te TO GET Ti. This will be both spatially variable across cells, due to soil type and textural qualities, and temporally variable due to varied occurrences and intensities of human disturbance.

SM_SWITCH = IF AVAILABLE_SOIL_MOISTURE _ SM_MIN THEN 1 ELSE 0

```
{THIS SHOULD MAKE THE DECREASE WORK OFF OF THE "SM_ADJUSTED_AE" WHEN
SM_STORAGE REACHES OR FALLS BELOW THE ALLOWED MINIMUM IN "SM_MIN"}
SM_TEMPERATURE = SM_TEMP_CELSIUS+((CV_ELEVATION-SM_BASE_ELEVATION) *.01) *
SM_SLOPE_RECLASS*SM_ASPECT_RECLASS
```

**DOCUMENT:** Units = degrees Celsius

```
SM_TEMP_CELSIUS = ((SM_BASE_TEMPERATURE-32.0)*5.0)/9.0
```

**DOCUMENT:** Units = degrees celsius

```
SM_Veg_index_test = ((CVP_SAGEBRUSH/17.5)+
(CVP_AGROPYRON/88.5)+
(CVP_FORB/7.1)) / 3.0
LEAF_AREA_INDEX = GRAPH(SM_Veg_index_test)
(0.00, 0.00), (0.0526, 0.03), (0.105, 0.075), (0.158, 0.17), (0.211, 0.38), (0.263, 0.735), (0.316, 0.905),
(0.368, 0.955), (0.421, 0.985), (0.474, 1.00), (0.526, 1.00), (0.579, 1.00), (0.632, 1.00), (0.684, 1.00),
(0.737, 1.00), (0.789, 1.00), (0.842, 1.00), (0.895, 1.00), (0.947, 1.00), (1.00, 1.00)
SM_BASE_TEMPERATURE = GRAPH(CV_WEEK)
(0.00, 27.5), (4.25, 34.0), (8.50, 42.0), (12.8, 50.5), (17.0, 58.5), (21.2, 64.4), (25.5, 71.0), (29.8, 68.6),
(34.0, 61.3), (38.2, 50.5), (42.5, 37.4), (46.8, 31.5), (51.0, 27.5)
```

**DOCUMENT:** Units = degrees fahrenheit

```
SM_INFILTRATION_RATE = GRAPH(SM_AVAILABLE_SOIL_WATER)
(0.00, 1.00), (11.1, 0.865), (22.2, 0.74), (33.3, 0.635), (44.4, 0.51), (55.6, 0.405), (66.7, 0.32), (77.8, 0.205),
(88.9, 0.085), (100, 0.00)
SM_MELT_VS_TEMP = GRAPH(SM_TEMPERATURE)
(0.00, 0.00), (1.50, 0.02), (3.00, 0.075), (4.50, 0.17), (6.00, 0.345), (7.50, 0.765), (9.00, 0.9), (10.5, 0.97),
(12.0, 0.99), (13.5, 1.00), (15.0, 1.00)
SM_PERCENT_LOSS_FROM_STORAGE = GRAPH(SM_AVAILABLE_SOIL_WATER)
(0.00, 0.28), (5.00, 0.3), (10.0, 0.325), (15.0, 0.375), (20.0, 0.455), (25.0, 0.585), (30.0, 0.765), (35.0, 0.88),
(40.0, 0.94), (45.0, 0.975), (50.0, 0.985), (55.0, 1.00), (60.0, 1.00), (65.0, 1.00), (70.0, 1.00), (75.0, 1.00),
(80.0, 1.00), (85.0, 1.00), (90.0, 1.00), (95.0, 1.00), (100, 1.00)
SM_RAINFALL_MM = GRAPH(CV_WEEK)
(0.00, 28.0), (4.25, 20.0), (8.50, 12.0), (12.8, 11.0), (17.0, 14.0), (21.2, 16.0), (25.5, 5.00), (29.8, 5.00),
(34.0, 11.0), (38.2, 14.0), (42.5, 25.0), (46.8, 27.0), (51.0, 28.0)
```

**DOCUMENT:** Units = millimeters of water per month

# Appendix D: GIS Input Maps

Each stock involved in the model must be initialized for time step zero. Raster maps processed with the GRASS GIS were used to initialize these variables. Model variables are listed below with a description of the GRASS maps used for initialization, some of which are included here.

## AG_COVER (Agropyron Cover)

*Agropyron* grass cover up to 50 percent. Other grasses were inconsequential in total grass cover (Figure D1).

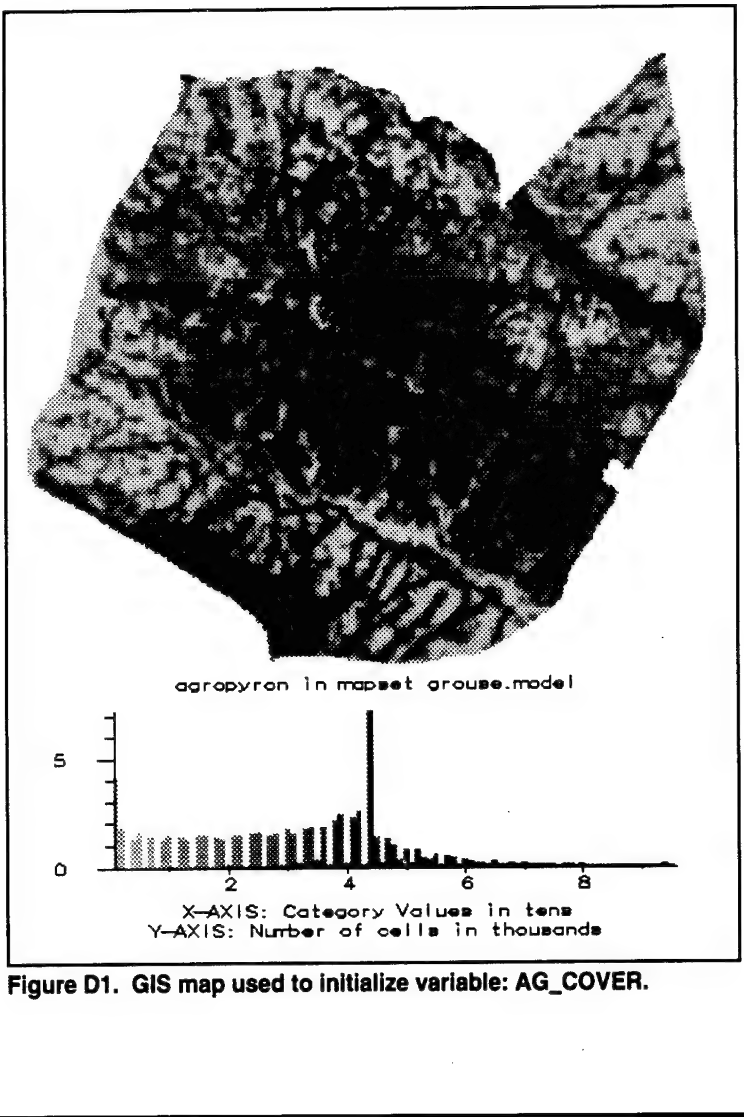

agropyron in mapset grouse.model
5
0
2
4
6
8
X-AXIS: Category Values in tens
Y-AXIS: Number of cells in thousands

Figure D1. GIS map used to initialize variable: AG_COVER.

## AVAILABLE_SOIL_MOISTURE

Units = millimeters (mm of water stored in the top 15 cm)

This storage used the units of millimeters of water within the rooting zone, assumed to be a depth of 15cm. The percent water available from the soil equaled the storage divided by 150 (mm).

## CV_ASPECT (Common Variable-Aspect)

The vegetation submodels were calibrated to accept zero to 364 degrees, computed directly from the elevation map that provided the CV_ELEVATION data. Slope direction was computed for each cell based on its nearest neighbors and was given in degrees from the east; values increased in a counterclockwise direction (Figure D2).

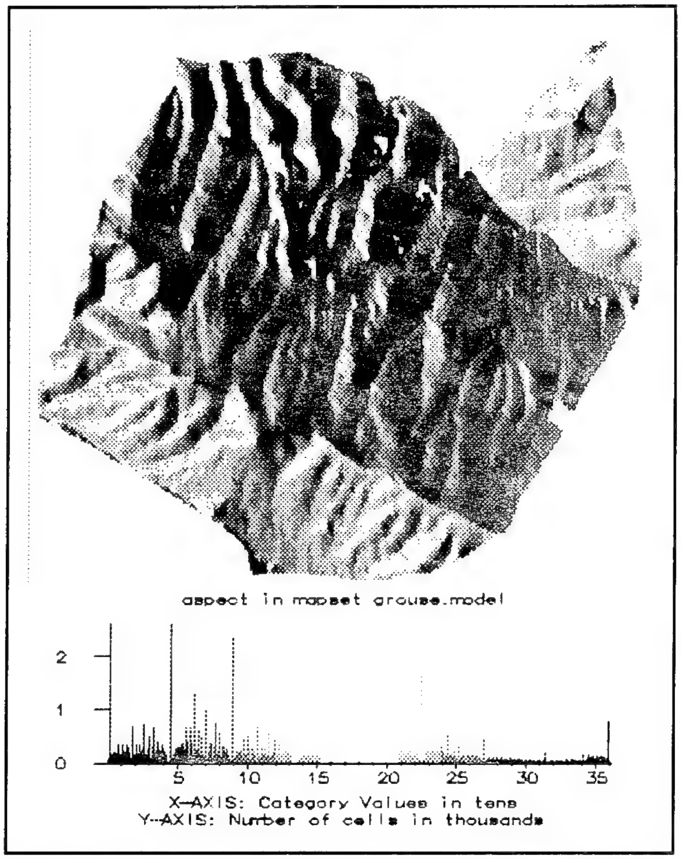

aspect in mapset grouse.model
2
1
0
5
10
15
20
25
30
35
X-AXIS: Category Values in tens
Y-AXIS: Number of cells in thousands

Figure D2. GIS map used to initialize variable: CV_ASPECT.

## CV_ELEVATION (Common Variable-Elevation)

Units were expressed in meters above sea level (Figure D3).

This variable was computed directly from the elevation map that provided the CV_ELEVATION data. Slope direction was computed for each cell based on its nearest neighbors and was given in degrees from the east; values increased in a counterclockwise direction.

## CV_ENCAMP_HR (Common Variable-Encampment Hours)

Units were expressed in personnel encampment hours per 100 hectares per exercise.

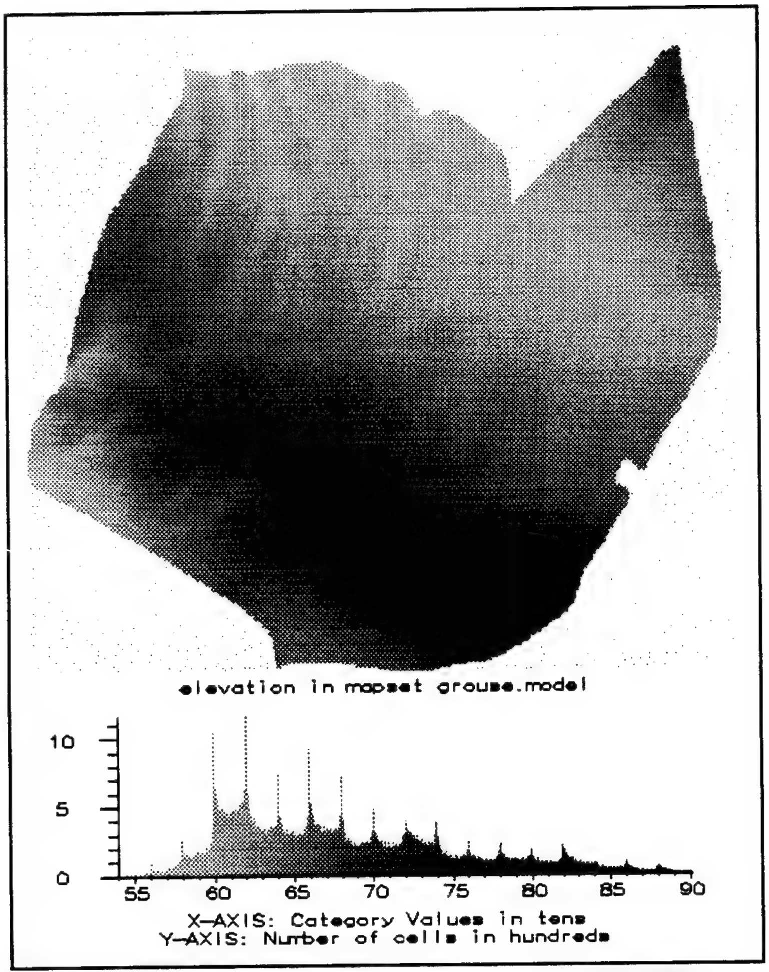

elevation in mapset grouse.model
10
5
0
55
60
65
70
75
80
85
90
X-AXIS: Category Values in tens
Y-AXIS: Number of cells in hundreds

Figure D3. GIS map used to initialize variable: CV_ELEVATION.

## CV_LEK_DIST (Common Variable-Distance From Lek)

Units were expressed in meters from the nearest lek.

A single lek site was situated toward the northeast corner of the study area. This map provided meter distances from each cell to the lek and was used to direct Grouse with respect to lek direction.

## CV_NOISE_INDEX

Index values between zero and 1 represent relative noise intensities. This was generated in the GIS using the CV_TRAK_VEH_HR map. Noise at any given location is an inverse weighted function of the tank densities in the surrounding area (Figure D4).

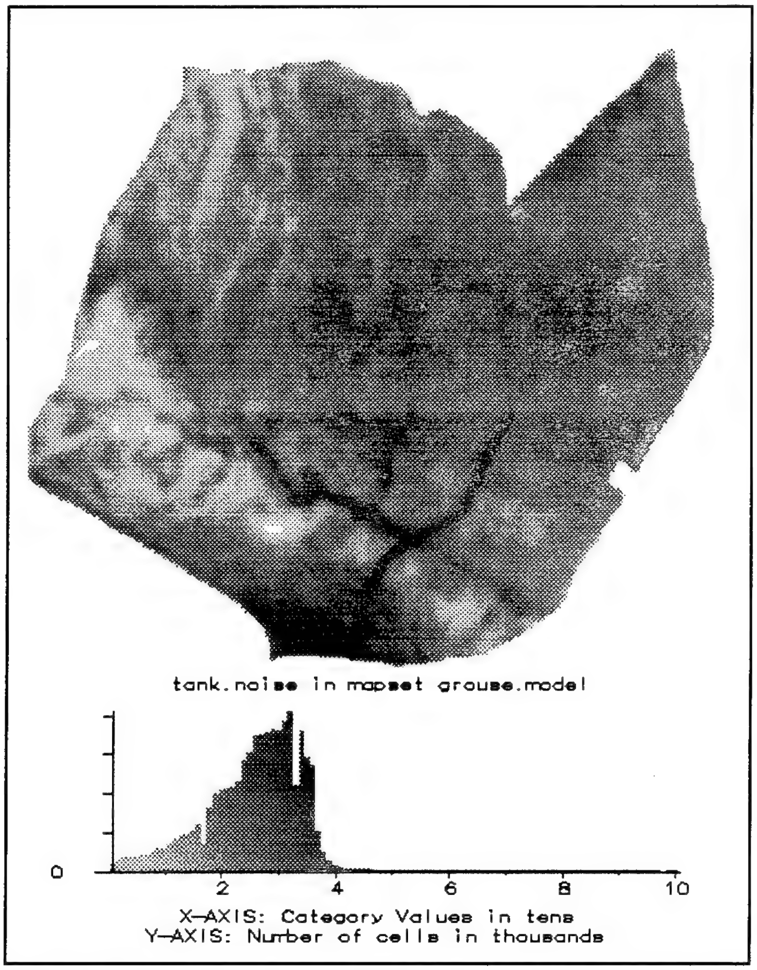

tank.noise in mapset grouse.model
0
2
4
6
8
10
X-AXIS: Category Values in tens
Y-AXIS: Number of cells in thousands

Figure D4. GIS map used to initialize variable: CVH_NOISE_INDEX.

## CV_SLOPE (Common Variable-Slope)

Slope was computed directly from the elevation map as well. Values were given in degrees from the horizontal (Figure D5).

## CV_WATER_HOLDING_CAPACITY (Common Variable-Water Holding Capacity of Soil)

This variable contained the measurement of millimeters of water held in the first 15 cm of soil with water at 100 percent saturation.

## CV_WETNESS_INDEX (Common Variable-Wetness Index)

This variable came from GRASS as a constant for each grid cell. It equaled [ln (A / TAN S)] where A = upslope area and S = slope. This value was set at 1.0. In other words, this fictitious grid cell had a value of 1.0.

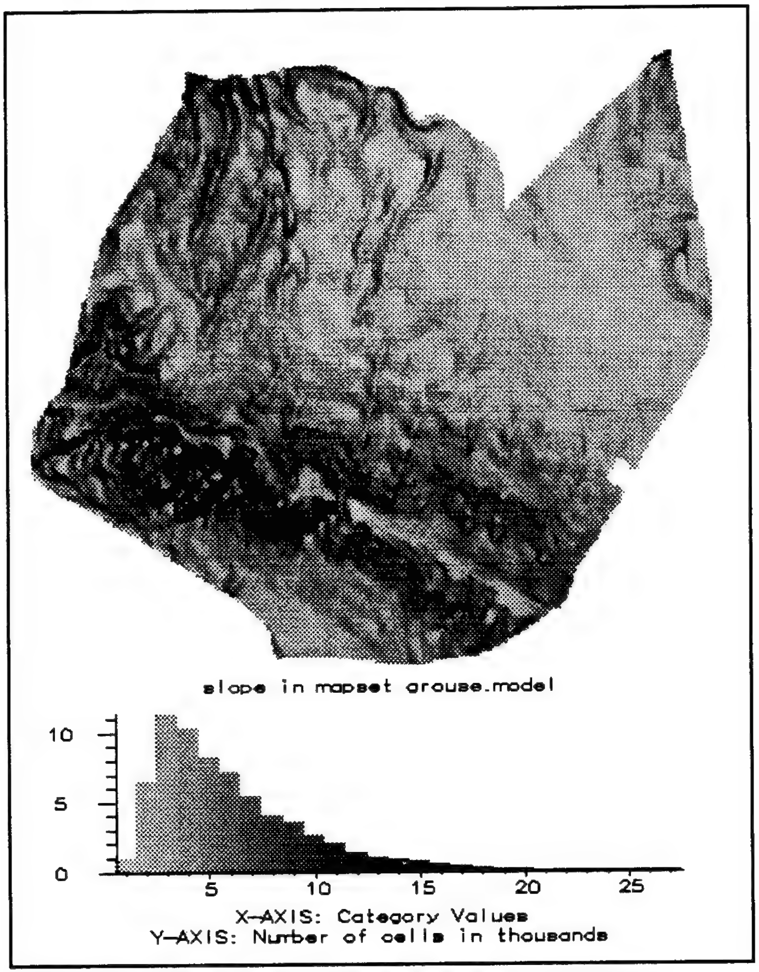

slope in mapset grouse.model
10
5
0
5
10
15
20
25
X-AXIS: Category Values
Y-AXIS: Number of cells in thousands

**Figure D5. GIS map used to initialize variable: CV_SLOPE.**

This was spatially variable across cells, not temporally variable due to the short duration being modeled.

### CV_TROOP_HR (Common Variable-Troop Hours)

Units = active troop hours per 100 hectares per exercise

### CV_UTRAK_VEH_HR (Common Variable-Untracked Vehicle Hours)

Units = moving untracked vehicle hours per 100 hectares per exercise

### CV_TRAK_VEH_HR (Common Variable-Tracked Vehicle Hours)

Units = moving tracked vehicle hours per 100 hectares per exercise

These maps (Figure D6) were generated to provide an estimate of the human impacts of a "typical" training event. In keeping with the approach and intent of this effort, the process demonstrates significant capabilities, but will require significant ground-truth and assistance from the target installation to attain a defensible accuracy.

Appendix B lists the GRASS analysis steps used to generate this map.

### F_SG_ADULTS (Female Sage Grouse Adults)

This map randomly placed 200 female Sage Grouse at the heavier densities of sagebrush throughout the study area.

### FIRE_DAMAGE

1 => There is a fire; 0 => No fire

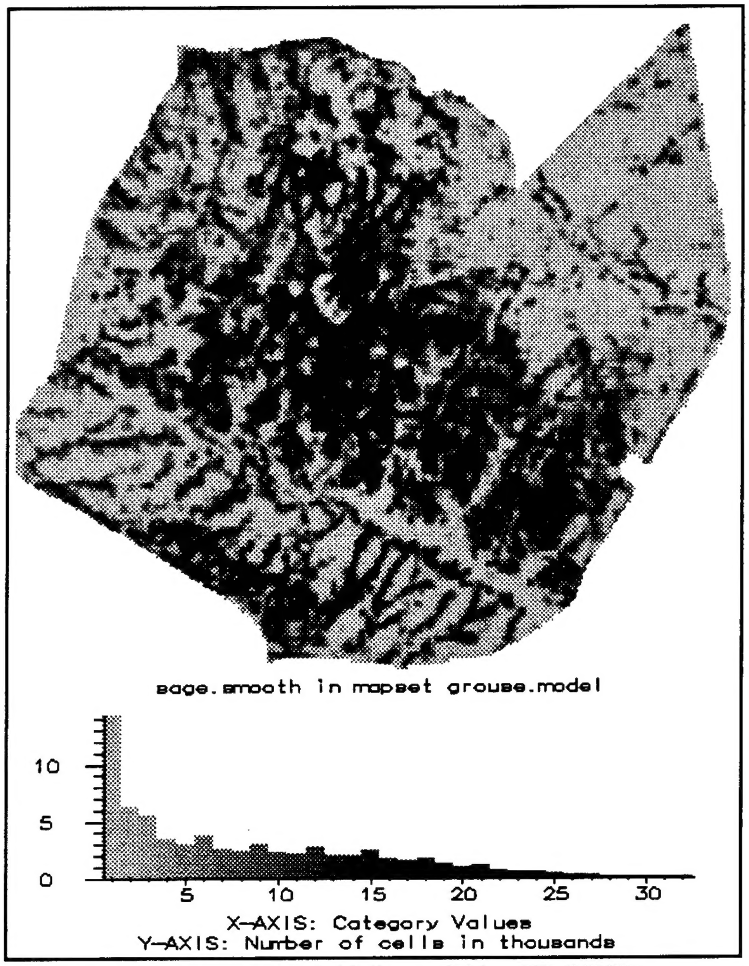

sage.smooth in mapset grouse.model
10
5
0
5
10
15
20
25
30
X-AXIS: Category Values
Y-AXIS: Number of cells in thousands

**Figure D6. GIS map used to initialize variable: SB_COVER.**

## ON_MAP (A 1/0 map Indicating Which Cells Were Part of the Study Area)

The model operated within a rectangular study area. To minimize processing time, it was given information regarding whether a given cell existed within the modeled region.

## SB_COVER (Common Variable-Sage Brush Cover)

Units = % cover (e.g., 25 = 25 percent)

Figure D7 includes both Wyoming big sagebrush and Threetip sagebrush.

These maps were created through an analysis of a June 1989 SPOT satellite image of the study area.

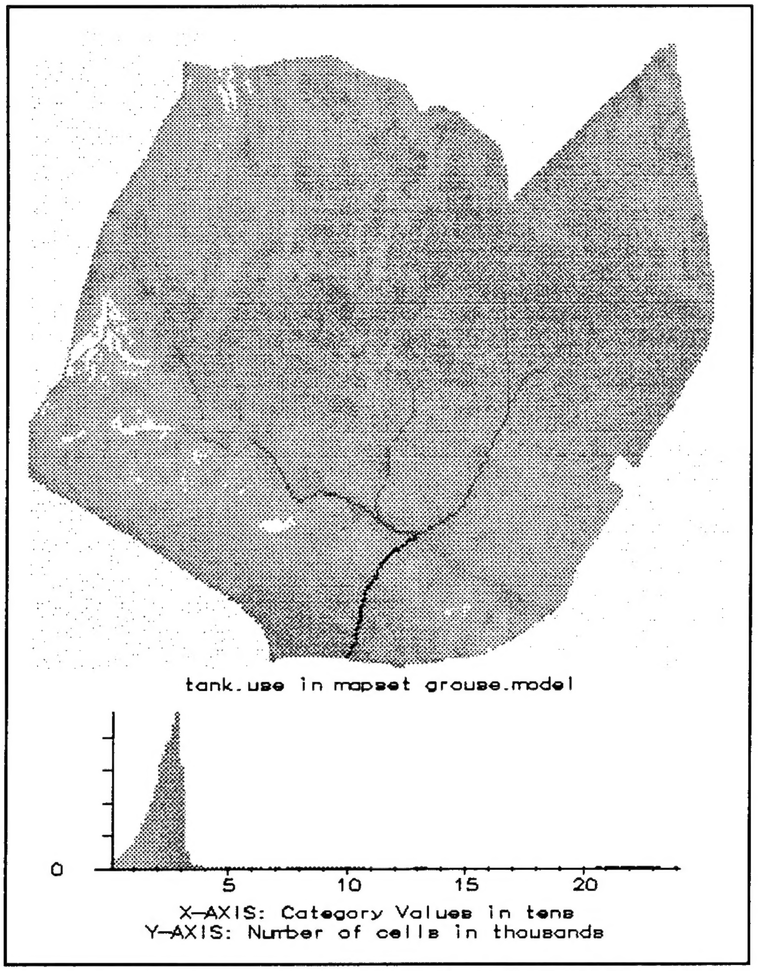

tank.use in mapset grouse.model
0
5
10
15
20
X-AXIS: Category Values in tens
Y-AXIS: Number of cells in thousands

**Figure D7. GIS map used to initialize variable: CV_TRAK_VEH_HR.**

image date: 8 June 1989
image time: 19:08:38
sun azimuth: +152.4
sun elevation: 64.0

This image was processed using a standard normalized vegetative index (NDVI) approach yielding results that correlated roughly with ground vegetation cover. Map values were normalized to reflect overall percent land cover values reported for the area.

# USACERL Distribution

Chief of Engineers
ATTN: CECPW-IM-LH (2)
ATTN: CECPW-IM-LP (2)
ATTN: CECC-R
ATTN: CERD-L
ATTN: DAEN-ZCI-P (2)
ATTN: CECW-PF
ATTN: CECW-EP-S
ATTN: CECW-RE
ATTN: CEIM-P
ATTN: CECW-OR

US Army Engineer District
ATTN: Chief, Regulatory Functions
Buffalo 14207
Norfolk 23510
Huntington 25701
Wilmington 28402
Charleston 29402
St. Paul 55101
Chicago 60606
Little Rock 72203
Galveston 77550
Albuquerque 87103
Los Angeles 90053
San Francisco 94105
Sacramento 95814
Portland 97208
Seattle 99362
New England 02254
ATTN: CENED-OD-R
New York 10278
ATTN: CENAN
Philadelphia 19106
ATTN: CENAP
Baltimore 21203
ATTN: CENAB-OP-R
Savannah 31402
ATTN: CESAS-OP-FP
Jacksonville 32232
ATTN: CESAJ-R-RD
Mobile 36628
ATTN: CESAM-OP-S
Nashville 37202
ATTN: CEORN
Memphis 38103
ATTN: CELMM-CO-R
Vicksburg 39180
ATTN: CELMV
Louisville 40201
ATTN: CEORL
Detroit 48226
ATTN: CENCE-CO
Rock Island 61204
ATTN: CENCR
St. Louis 63101
ATTN: CELMS-OD-F
Kansas City 64106
ATTN: Chief, Permits Section
Omaha 68102
ATTN: CEMRO-OP
New Orleans 71060
ATTN: Chief, Permits Section
Tulsa 74121
ATTN: CEWST
Fort Worth 76102
ATTN: CESWF-OD-O

US Army Engr Divisions
North Central 60606
ATTN: CENCD-CO
Southwest 75242
ATTN: CESWD-CO-R

US Military Academy 10996
ATTN: Dept of Geo & Envr Engr (2)
ATTN: Natural Resources Branch

Aberdeen Proving Ground, MD 21005
ATTN: ISC-TECOM
ATTN: HSHB-CI

US EPA Research Lab 97333

Yakima Training Center 98901

USATHAMA 21010
ATTN: CETHA-RM-I
ATTN: AMXTH-RM
ATTN: CETHA-IR-S

CECPW 22060
ATTN: CECPW-FN
ATTN: CECPW-SP
ATTN: CECPW-FM-A

Bureau of Land Management
WASH DC 20240
Fort Collins 80526
Denver 80225

US Dept of Commerce 20233

Army National Guard 20310
ATTN: NGB-AREC

US Army Concepts Analysis Agency 20814

FBI Academy 22135

USA Foreign Science Tech Ctr 22901

Naval Oceanographic Office 39522

Redstone Arsenal 35898
ATTN: AMSMI-RA-EH-MP-PC

CEWES
ATTN: CEWES-IM-DA 39181
ATTN: CADD Center 39180

Wright-Patterson AFB 45433
ATTN: NSBIT AL/OEBN

Michigan Dept of Military Affairs 48913

Twin Cities Army Ammo Plant (2) 55112

Camp Ripley 56345
ATTN: Ofc of Archeology & Engr (2)

5th Inf. Fort Polk 71459
ATTN: Envr & Nat Rsrcs Mgmt Div (2)

US Army Materiel Cmd 61299
ATTN: AMXEN-U (2)

US Army Europe
HQ USAREUR 09403
ATTN: AEAEN-FE-E (2)

Texas Army Nat'l Guard 78763

Lone Star Army Ammo Plant 75505

Red River Army Depot
ATTN SDSRR-GB 75507

Dugway Proving Ground 84122
ATTN: DPG-EN-E (2)

Yuma Proving Ground 85365
ATTN: ATEYP-ES-E

White Sands Missile Range
ATTN: STEWS-ES-E 88002

Envr Response & Info Ctr
ATTN: ENVR-EP 20310

Nat'l Geophysical Data Ctr
ATTN: Code E-GCI 80303

Hohenfels Training Area 09173
ATTN: AETTH-DEH
ATTN: AETTH-DEH-ENV-APO

US Army Forts
Fort Belvoir, VA 22060
ATTN: CEETL-CL-GC (2)
ATTN: CETEC-CA-D
ATTN: AMSEL-RD-NV-VMD-TST
ATTN: Envr & Nat Res Div

Fort Monroe, VA 23651
ATTN: ATBO-GE
ATTN: ATEN-FN
Fort Drum 13603
ATTN: AFZS-EH-E
Fort Jackson 29207
ATTN: ATCJ-EHN
Fort Gillem 30050
ATTN: FCEN-CED-E
Fort Gordon 30905
ATTN: ATZH-DIE (2)
Fort Stewart 31314
ATTN: AFZP-DEN-W
Fort Benning 31905
ATTN: Nat. Resource Mgmt Div (2)
Fort McClellan 36205
ATTN: ATZN-FEE
Fort Rucker 36362
ATTN: ATZQ-EH
Fort Knox 40121
ATTN: ATZK-EHE
Fort Campbell 42223
ATTN: AFZH-DEH
Fort Benjamin Harrison 46216
ATTN: ATZI-ISP (2)
Fort McCoy 54656
ATTN: AFZR-DEN
Fort Riley 66442
ATTN: AFZN-DE-N (2)
Fort Chaffee 72905
ATTN: ATZR-ZFE (2)
Fort Sill 73503
ATTN: Fish & Wildlife Br (2)
Fort Leonard Wood 65473
ATTN: ATZT-DEH-EE
Fort Dix 08640
ATTN: ATZD-EHN
Fort Eustis 23604
ATTN: Ranges & Targets Dir
Fort Worth 76115
ATTN: Cartographic Ctr (2)
Fort Hood 76544
ATTN: AFZF-DE-ENV
Fort Bliss 79916
ATTN: ATZC-DEH-E
Fort Carson 80913
ATTN: AFZC-ECM-NR
Fort Huachuca 85613
ATTN: ATZS-EHB
Fort Irwin 92310
ATTN: AFZJ-EH
Fort Lewis 98433
ATTN: AFZH-DEQ
ATTN: ATZH-EHQ
Fort Richardson 99505
ATTN: DEH
Fort Bragg 28307
ATTN: DEH

National Weather Service 20910

US Geological Survey 22092

Pine Bluff Arsenal 17602
ATTN: SMCPB-EMB

US Army Cold Regions Research & Engr Lab
ATTN: CECRL-IS 03755

NASA/SSC/STL 39529

Defense Technical Info Center
ATTN: DTIC-FAB (2)

146
8/95